A BZ Media Publication

SD Times

SOFTWARE DEVELOPMENT

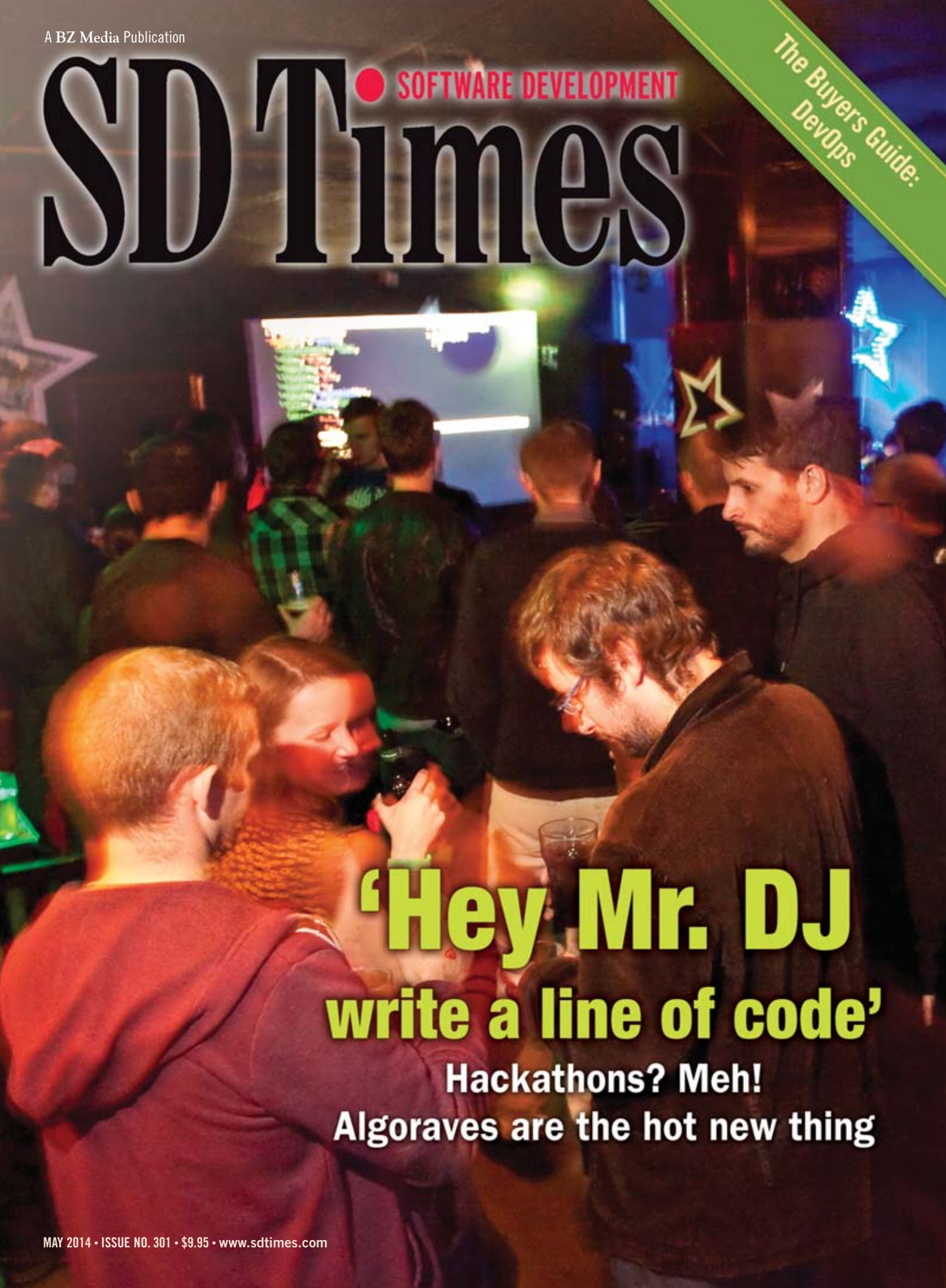
MAY 2014 • ISSUE NO. 301 • $9.95 • www.sdtimes.com

A BZ Media Publication

SD Times

SOFTWARE DEVELOPMENT

The Buyer's Guide: DevOps



'Hey Mr. DJ write a line of code'

Hackathons? Meh! Algoraves are the hot new thing

MAY 2014 • ISSUE NO. 301 • $9.95 • www.sdtimes.com

Data Quality Tools for Developers

A better way to build in data verification

Since 1985, Melissa Data has provided the tools developers need to enhance databases with clean, correct, and current contact data. Our powerful, yet affordable APIs and Cloud services provide maximum flexibility and ease of integration across industry-standard technologies, including .NET, Java, C, and C++. Build in a solid framework for data quality and protect your investments in data warehousing, business intelligence, and CRM.

- **Verify international addresses for over 240 countries**
- **Enhance contact data with phone numbers and geocodes**
- **Find, match, and eliminate duplicate records**
- **Sample source code for rapid application development**
- **Free trials with 120-day ROI guarantee**

Address Verification

Phone Verification

Email Verification

Geocoding

Matching/ Dedupe

Change of Address

Visit Us at TechEd Booth #1934

Melissa Data.
Architecting data quality success.

MELISSA DATA®

www.MelissaData.com 1-800-MELISSA

No risk, no reward? ...bullsh*t!

Risk is the enemy of reward. For nearly 20 years, Seapine has helped companies like yours mitigate risk in product development. Whether it's schedule slippage, cost overruns, product safety and compliance, or product quality, our tools will help you reduce risk and reap the rewards.

Seapine's time-proven solution, TestTrack, helps take the risk out of product development and gives you:

Visibility. Know where you are on schedule, the current level of quality, the potential impact of a change, the source of a problem, coverage of tests against requirements… you get the picture.

End-to-End Lifecycle Management. From requirements and user stories through test cases, results, and resolutions, TestTrack manages it all in one centralized repository for full backward and forward traceability.

Flexibility. Multiple workflows and methodologies, hosting in the cloud or on-premise, web or desktop clients, no solution is more flexible.

Don't accept risk in product development. Manage it with Seapine TestTrack and reap the rewards.

Learn how to take the risk out of product develoment.

www.seapine.com/less-risk

Seapine Software™

Contents

ISSUE 301 • MAY 2014

FROM THE EDITORS

NEWS

COLUMNS

Cover photo by www.underyourskinphoto.co.uk

FEATURES

BUYERS GUIDE

SPECIAL REPORT

Software Development Times (ISSN 1528-1965) is published 12 times per year by BZ Media LLC, 225 Broadhollow Road, Suite 211, Melville, NY 11747. Periodicals postage paid at Huntington Station, NY, and additional offices. SD Times is a registered trademark of BZ Media LLC. The price of a one-year subscription is US$179 for subscribers in the U.S., $189 in Canada, $229 elsewhere. POSTMASTER: Send address changes to SD Times, 225 Broadhollow Road, Suite 211, Melville, NY 11747. SD Times subscriber services may be reached at subscriptions@bzmedia.com.

Attend the Largest Dedicated Android Development Conference in the Universe!

Check out these NEW classes:

An Introduction to Building Enterprise-Secure Android Apps
G. Blake Meike

Android as the New Standard Embedded OS
Karim Yaghmour

Android Security: New Threats, New Capabilities
Jonathan Levin

Connect Your App to the Real World with Google Services Location API
Jim Wilson

Creating Composite Views in Android
James Harmon

Customizing Android for Fun and Profit, Parts I and II
Dave Smith

Developing Android Bluetooth Smart Ready Apps
Vincent Gao

Five Strategies to Achieve Continuous Unattended Testing
Uzi Eilon

Automotive Android
Godfrey Nolan

Git for the Android Developer, Parts I and II
Tony Hillerson

Gradle: From User to Addict
Jake Ouellette

Home Sweet Home: Widgets and Live Wallpapers
Elizabeth Mezias

Push Message Primer
Mark Wickham

The Art of Building Reusable UI, Parts I and II
Stephen Barnes

Top 5 Android Performance Tips
Godfrey Nolan

Who's Afraid of ContentValues?
Ian Darwin

Android Networking for the Enterprise
G. Blake Meike

Understanding Android Studio and Its Role in the Android Developer Toolset
Jim Wilson

A Totally Different View: Under the Hood with Android and Linux
Jonathan Levin

Leveraging Fully Customized ViewGroups From the Ground Up
James Baca

OpenGL-ES 3.0 and Beyond—Modern OpenGL Programming
Ron Fosner

...and these conference favorites:

Android Emulator Myths... Busted
Luis de la Rosa

Battle-Tested Patterns in Android Concurrency, Parts I and II
Doug Stevenson

Bootstrapping Android Development
Donn Felker

Dive into Android Fragments
Donn Felker

Talk to Your Toaster Part I: Bluetooth in Your Application
Dave Smith

Talk to Your Toaster Part II: Developing USB Accessories
Dave Smith

The Action Bar, Front to Back
Mark Murphy

Embedded Android Tutorial
Karim Yaghmour

More classes online at AnDevCon.com/classes.html




Register TODAY at www.AnDevCon.com

A BZ Media Event #AnDevCon

SD Times SOFTWARE DEVELOPMENT

EDITORIAL

EDITOR-IN-CHIEF David Rubinstein
631-421-4158 x105 • drubinstein@bzmedia.com

SENIOR EDITOR Alex Handy • ahandy@bzmedia.com

COPY EDITOR Adam LoBelia • alobelia@bzmedia.com

SENIOR ART DIRECTOR Mara Leonardi • mleonardi@bzmedia.com

SOCIAL MEDIA AND ONLINE EDITORS Rob Marvin • rmarvin@bzmedia.com
Christina Mulligan • cmulligan@bzmedia.com

COLUMNIST Larry O'Brien

CONTRIBUTING WRITERS Alyson Behr, Patrick Hynds, G. Arnold Koch, Lisa L. Morgan, Dave Smith, Alexandra Weber Morales

CONTRIBUTING ANALYSTS Rob Enderle, Mike Gilpin, Mike Gualtieri, Jeffrey Hammond, Al Hilwa

CUSTOMER SERVICE

SUBSCRIPTIONS subscriptions@bzmedia.com

ADVERTISING TRAFFIC Shauna Koehler
631-421-4158 x112 • adbox@bzmedia.com

LIST SERVICES Shauna Koehler
631-421-4158 x112 • skoehler@bzmedia.com

REPRINTS Stacy Burris
631-421-4158 x108 • sburris@bzmedia.com

ACCOUNTING Viena Ludewig
631-421-4158 x110 • vludewig@bzmedia.com

OTHER QUESTIONS 631-421-4158 x129• mchyriwski@bzmedia.com

ADVERTISING SALES

PUBLISHER David Lyman
978-465-2351 • dlyman@bzmedia.com

EUROPE, MIDDLE EAST WESTERN ASIA David Lyman
978-465-2351 • dlyman@bzmedia.com

WESTERN U.S., WESTERN CANADA, EASTERN ASIA AUSTRALIA, INDIA Paula F. Miller
925-831-3803 • pmiller@bzmedia.com

EASTERN U.S. EASTERN CANADA Jonathan Sawyer
603-924-4489 • jsawyer@bzmedia.com

PUBLISHING DIRECTOR Ted Bahr
631-421-4158 x101 • ted@bzmedia.com

BZ Media

PRESIDENT Ted Bahr

FOUNDING EDITOR Alan Zeichick

BZ MEDIA LLC
225 Broadhollow Road, Suite 211
Melville, NY 11747
TEL 631-421-4158
FAX 631-421-4130
www.bzmedia.com
info@bzmedia.com

THE MOST ADVANCED TECHNOLOGY FOR BUILDING RICH INTERNET APPLICATIONS

SmartGWT makes it simple to build powerful, high-productivity web applications with the skills you have today.

Write high-productivity UIs in Java

- Use existing skills & standard Java IDEs to write modern web & mobile apps
- Java code runs as JavaScript in the browser, so UI responds instantly
- Build faster & better with the broadest, deepest UI component set available

Leverage built-in mobile support

- UI components automatically change appearance & behavior for mobile
- Unified touch and mouse event handling - your custom logic "just works" on mobile
- Single codebase instead of separate mobile and desktop UIs

Accelerate development with deep server-side integration

- Generate UIs directly from your server-side data model, and customize without losing auto-generation
- Declare data validation rules once and get both client and server enforcement
- Built-in support for JPA/Hibernate, Spring & SQL makes it easy to connect your UI to data

Get everything you need from an enterprise-class technology

- Mature, advanced, professionally backed solution
- Rock solid: 10s of thousands of automated tests ensure consistent, reliable releases
- Consulting, SLA-based support, and a community of experts

Isomorphic™ SOFTWARE

Try it now at www.SmartClient.com/SmartGWT

Bleeding hearts are not so bad

Did you patch OpenSSL? If not, why are you reading this? Go patch!

For everyone else, go ahead and calm down. Let's be reasonable here. The danger has passed for now. The real danger here is not even Heartbleed, frankly.

We're not saying Heartbleed wasn't dangerous. What we are saying, however, is that the real problem wasn't the ability to read data from OpenSSL's RAM cache. The real problem was the fact that an entire world of Web-based computing relies on a project that amounts to little more than four or five people who aren't paid very much.

The industry needs to support OpenSSL with more resources, more eyeballs, and more developers. As of this writing, the OpenSSL development team consists of 11 people. We've heard that only about two of them were actively maintaining the project. This is exactly why the OpenBSD folks decided to turn their gaze upon OpenSSL. The expertise of the OpenBSD team will shine a lot of light into the narrow corners of the OpenSSL project.

But we'll need more than them. Everyone needs to tuck in and fix OpenSSL. (Perhaps "fix" is not the right phrase for what needs to be done, however. Perhaps the proper word is "advance.")

A lot of ire came out during the weeks following Heartbleed, and much of it was aimed at the way OpenSSL behaves and how it is designed. One spurious criticism was that C and C++ should not be used for security-critical infrastructure.

That's no solution, and we've all known how to develop securely in C for many years. Often exploits come from lazy coding, not the fact that the language does not support memory management.

A far more coherent argument reasons that OpenSSL is extremely old, TLS is obtuse, and when combined, these two elements make for a terrible development experience.

It is the frustration from dealing with inhospitable programming paradigms that leads to developers not implementing encryption properly, or to their sloppy coding.

But there is no excuse for this particular bit of sloppy coding. The Heartbleed attack was a serious wake-up call for the industry. We're sincerely hopeful that the open-source community, like OpenBSD, can pitch in and advance OpenSSL so that it can catch up to the rest of the world in terms of security, ease of use, and reliability. But that's the thing about open source, isn't it? If one of the cornerstones of our industry crumbles, it's our own fault. ∎

'Because we can' isn't good enough

This seems to have come straight from the "I guess we really didn't think it through" department:

As reported on sdtimes.com, developer Sander Veenhof created an app that lets people know about "privacy intrusions caused by surveillance cameras." (Quotation marks are our own, for emphasis.) Apparently, Google Glass wearers can see green "safe" zones where no cameras are around, and red "hot" zones in which they are being watched.

This, apparently, is to protect Glass wearers from an invasion of their privacy in areas covered by closed-circuit cameras. (We're not exactly sure how this is classified as an invasion of privacy, but that's another matter entirely.)

So we ask... are we missing something here?

It seems to us that cameras are used to protect the properties to which they are attached. For instance, a jewelry store will have a camera inside to watch for burglars, and to capture their image in the event of a crime. So will a bank, and a convenience store. Some will even post a sign that says "closed-circuit cameras in operation" without really having them, to use as a deterrent to crime.

Now, imagine a burglar steals the Google Glass and has this app installed. He'll simply go to a store that does NOT have surveillance and rob that one.

Is this really what we want? Technology that can aid and abet people with criminal intentions? We certainly hope not.

Like the headline says: Just because we can create this cool software that can do certain things doesn't mean we ought to. While software in the vast majority of cases is created to improve our lives, it still too often is misused by malicious groups or individuals.

Technology is racing out of control, as very brilliant minds create all kinds of applications and devices simply because they can. Drones that deliver pizzas? Applications that wearers can use to avoid detection?

There needs to be some kind of industry oversight to slow this pace, to examine why things are being created and to what end they can be used, before we are simply left with a very bad outcome and the lament, "I guess we didn't really think it all the way through." ∎

SD Times ON THE WEB

A whole new meaning to 'end of life'

Windows XP and Internet Explorer 6 both had their plugs pulled in April, but even at the end they still had people using them. Microsoft wants people to get with the times, and to help spur a move away from the old, it has created Escape from XP, a game allowing players to blow away icons from Microsoft's XP era. "Some other old friends (think a certain paperclip helper) also show up along the way," reports Rob Marvin. If you're eager to see Office Assistant ruin things one last time, check it out here: sdt.bz/70051.

Who's worried about Oculus?

It's been a while since Facebook bought Oculus for a few billion. Those who chipped in for its Kickstarter were not too happy, but is this good news for developers in general? Not according to Alex Handy: "John Carmack asked specifically what people were worried about via his Twitter feed. My first take on why people are worried is purely related to him: After an acquisition of any kind, talent flees like rats off a sinking ship." You can read Alex's full take at sdt.bz/68997.

Finally, a way to wear JavaScript on your face

Google Glass development is advancing fast, and the latest breakthrough comes from two Ph.D. candidates. WearScript is an open-source JavaScript framework that aims to let people build apps for Glass as if they were any other kind of Web app. "The framework can utilize or repurpose any of Glass' native capabilities while also connecting the wearable to other input devices and hardware," reports Rob Marvin. You can see more for yourself at sdt.bz/68986.

Every breath you take
(I'll be encrypting you)

It's difficult to keep ahead of hackers when it comes to software security, but researchers may have found a new way to do it. "The encryption method is based on coupling functions, which are what allow the heart and lungs to exist independently while operating in sync," reports Rob Marvin. That's right: Signals between your heart and lungs (in a process known as system coupling) can be used to encrypt data. That heartbeat you have may be used to keep your data secure, if this method takes off. More is available at sdt.bz/69025.

The Top 5 GitHub projects from April

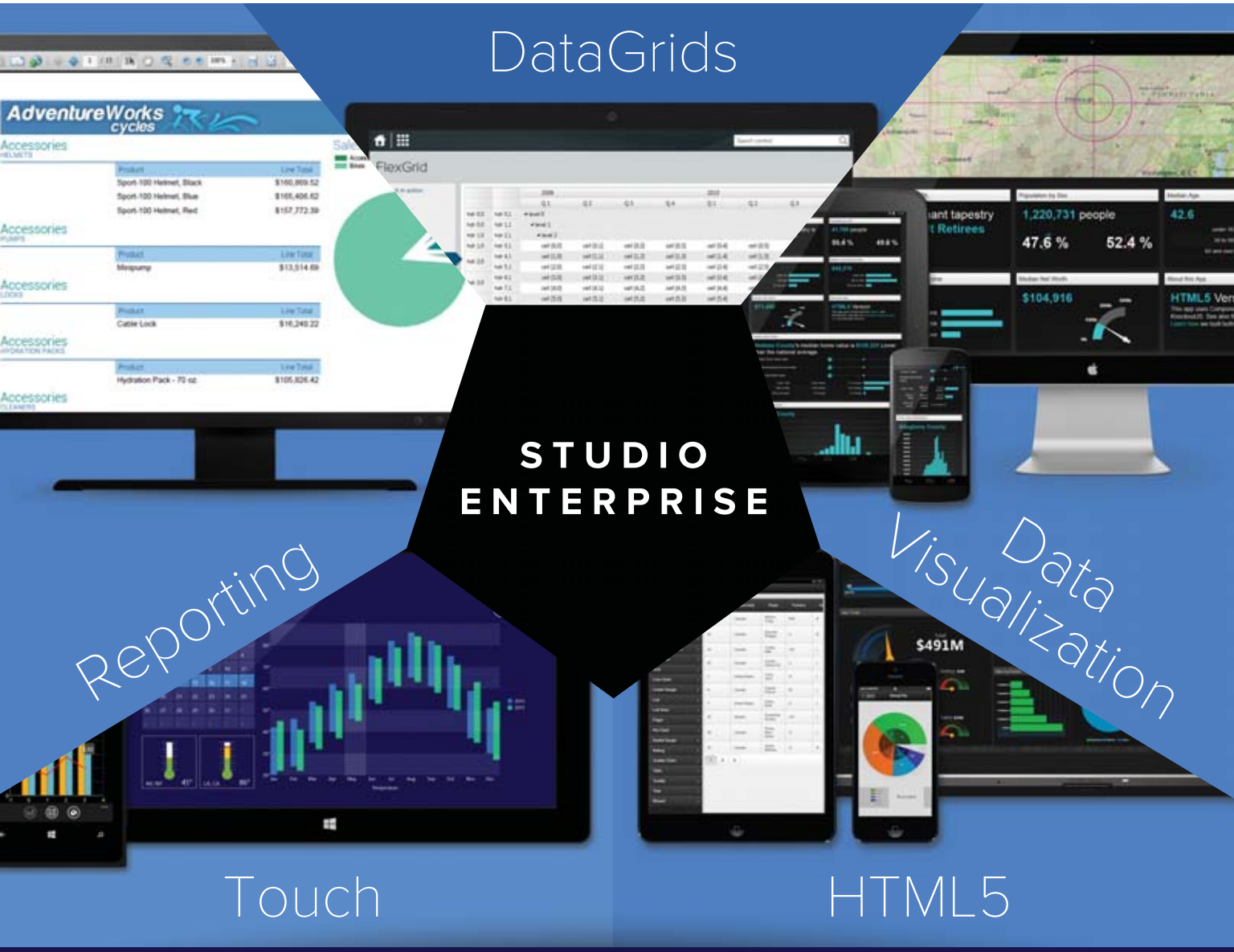
The world's largest project repository allows you to see what the most popular projects are. Here's what saw the most action in April:

1) **Slick**
2) **GitHub Cheat Sheet**
3) **Free Programming Books**
4) **2048**
5) **GitBook**

SD Times wants to hear from you.
Join us on LinkedIn and Facebook.

.NET TOOLS FOR DEV PROS

Whether you're building the most modern touch-enabled apps or maintaining and updating legacy applications, our flagship product, Studio Enterprise, helps to deliver rich, responsive, desktop and web apps on time and under budget.



New 2014 v1 Release!

ComponentOne®
a division of GrapeCity®

DOWNLOAD YOUR FREE TRIAL
▶ componentone.com/se

Microsoft unifies development platforms

Build conference introduces universal Windows apps, Windows Phone 8.1

BY ALEX HANDY

Keynote addresses at this year's Build conference focused on Windows desktops, Windows Phone and Windows Surface. These three platforms were unified from a development perspective, as Microsoft announced universal Windows apps during the keynote.

"With the Windows Phone 8.1 release, we brought the new Windows Runtime to phones. Now you can produce common apps across phones, PCs and tablets," said David Treadwell, corporate vice president at Microsoft. "To enable universal Windows apps, we've streamlined every phase of Windows development.

"You need to reach customers across multiple devices. Windows will help you do that. We know you've made a huge investment in your code and you need to carry that forward. Windows will help you do that. You need to be able to deliver apps across platforms. Windows will do that."

Users can ask Cortana to help with tasks, such as creating an appointment, setting an alarm, calling, texting, getting directions, and more.

Kevin Gallo, group program manager at Microsoft, showed off the mechanisms of universal Windows apps with a Visual Studio demonstration. The same code, whether C#, C/C++ or even now JavaScript, can be pushed into any of Microsoft's three platforms by sharing chunks of code, he said.

"You can very easily mark what code and assets you want to share," said Gallo. "Since I built this entire thing, I want to share the entire app. It's taking everything in my Windows 8.1 project—all the code, all the images, and even all my XAML—and moving it into my shared project. Like you, I have some third-party libraries. I used Json.NET here. We don't assume you're going to share, but you just move what you want to share and put it in the shared node."

The results of his work allowed Gallo to quickly transfer a Windows-based full-screen application into a slide-to-browse Windows Phone application.

Treadwell went on to explain that Microsoft Office is being transferred to Windows Phone by becoming a universal Windows app. Kirk Koenigsbauer, corporate vice president at Microsoft Office, demonstrated the advances the Office team has been experimenting with on non-desktop platforms.

"From a developer perspective, the

So, is it a universal platform or not?

Weeks after announcing universal Windows apps at the Build developer conference, Microsoft is seemingly fragmenting its platforms by killing Windows 8.1 while continuing to push Windows Phone 8.1.

In a TechNet article, Microsoft desktop and application virtualization consultant Steve Thomas announced an end of support and security patches for Windows 8.1, Windows 8.1 RT and Windows Server 2012 R2 in the next 30 days. After that point, only those who've installed the Windows 8.1 Update 1 or are still running Windows 8 will continue to receive updates.

At the same time, Microsoft released Windows Phone 8.1 and announced the availability of the Windows Phone 8.1 developer preview, promoting the building and testing of universal Windows apps. The move appears to sow a divide between a platform pushing forward and a platform in disarray, as customers who haven't yet updated to the error-ridden Windows 8.1 Update 1 are left in limbo between an obsolete OS and a currently nonexistent update.

The concept of universal Windows apps is a seamless developer application experience across desktops, tablets and smartphones. Yet without cohesiveness between Windows 8.1 and Windows Phone 8.1, it remains to be seen whether apps can be truly universal until the schism is resolved. The Windows and Windows Phone divisions should be acting in unison, not moving in opposite directions. ▮

—Rob Marvin

Streamlining development is important for universal Windows apps, says David Treadwell.

Windows Runtime has been a great place for us to build highly responsive apps," he said. "We're trying to bring forward the experiences from Office on Win32 to the modern platform itself."

Koenigsbauer then demonstrated how the listing of recently used documents are linked to Microsoft's Office 365 online offering, allowing desktops, phones and tablets to see the same recently used list.

Treadwell said that old applications can be ported forward to become universal Windows apps with minimal effort. A demonstration that involved porting old ADO.NET code to a Windows Tablet showed how even synchronous calls could be worked around.

Of clouds and Cortana

Elsewhere at the show, Microsoft demonstrated Cortana, its answer to Apple's Siri and Google's Now service. With the release of Windows Phone 8.1, said Treadwell, developers no longer need to define a grammar for interfacing with Cortana. Rather, Cortana can discern needs from its existing cloud-based database.

Cortana is based on the character from the Halo series of games.

Windows Phone 8.1 will also include dozens of new features for developers. New video-editing capabilities will be available. Developers will also have the ability to poll background triggers. These triggers will allow applications to respond to push notifications and geofencing.

Treadwell also announced the open-source availability of Microsoft's JavaScript interface library, WinJS. The project is now available on GitHub under an Apache 2.0 license. He said the project was also becoming cross-platform.

Microsoft also announced that it has been working on Roslyn, a next-generation compiler for Visual Basic and C#. This compiler is designed not to be so much as a black box as regular compilers are. The goal is to make the compiler behave more like an API.

Roslyn is a work in progress, but at Build, Microsoft announced that it would be made open source.

As for the future, the company plans to offer a version of Windows designed for embedded devices and the Internet of Things for free to developers. Devices with screens smaller than nine inches are also eligible to receive future Windows updates for free.

The Xbox One, on the other hand, is becoming part of the universal Windows apps pantheon, though no actual date for this functionality was given. In the future, though, developers will be able to deploy their desktop and phone apps onto the Xbox One using the universal Windows apps capabilities of Visual Studio.

A new version of Visual Studio supporting universal Windows apps is available from MSDN right now; however, this is only a release candidate for Visual Studio and not a final release.

Why Windows?

Microsoft CEO Satya Nadella addressed the crowd at Build by taking some prerecorded questions at the end of the keynote address. One of those questions caused him to ruminate on Microsoft's past, and its similarity to its future.

"If you think about what developers mean to us, it's pretty deep," he said. "We were a tools company before we were an office company before we were a Windows company. We are again in that era now. We're bringing the entirety of the Windows family from the Internet of Things to consoles to tablets to phones to PCs. We have that proliferation in ubiquitous computing in ambient intelligence. It's exciting times for us and developers in terms of opportunity to take your apps and bring them forth to Windows as it evolves."

When asked why someone should build for Windows, Nadella said, "You want to build for Windows because we are going to innovate with a challenger mindset. We're not trying to do just the next version of Windows; we're going to come at this by innovating in every dimension the software and hardware across the Windows family. We'll make progress with rapid pace." ▮

Photos courtesy of Microsoft

Satya Nadella explains Microsoft's plan to move Windows into the Internet of Things.

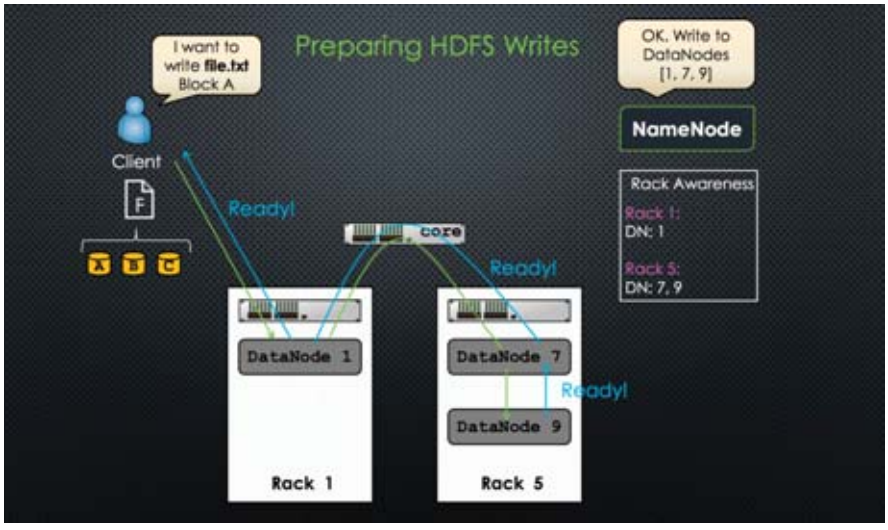
The 411 on Big Data TechCon

Conference focuses on gaining insights from fire hoses of information

BY DAVID RUBINSTEIN

CAMBRIDGE, MA — Need proof that the momentum around Big Data isn't waning any time soon? One would have needed to look no further than the packed Hyatt Regency ballroom here at last month's Big Data TechCon, where an introductory, full-day, hands-on, all-things-Hadoop tutorial drew a capacity crowd eager to take the plunge.

In that session, attendees were introduced to the core concepts of Hadoop, as well as the critical paths of HDFS, Map/Reduce and HBase. The basics of how to effectively write Pig and Hive scripts, and how to choose the correct use cases for Hadoop, were also covered.



A slide from "Hadoop: A One-Day, Hands-On Crash Course," taught by Sameer Farooqui.

So, as the Big Data newbies got an immersion in Hadoop, others were looking for insights into doing analytics against the data. After all, what good is collecting all that data if you can't gain a competitive edge for your organization by understanding what the data contains?

That is what Big Data TechCon is about. If you missed last month's event, you can analyze this: The next Big Data TechCon is set for Oct. 27-29 in Burlingame, Calif.

Big Data TechCon is produced by BZ Media, which publishes SD Times. ▮

Report: Collaborative development is on the rise

BY ROB MARVIN

Collaborative software development is experiencing a period of rapid growth, according to a new report from the Linux Foundation.

The foundation defined collaborative development as "software development that involves multiple individuals and companies, in many cases competing in the same industry, and in which the codebase is open source and a shared investment."

During the first week of March, an invitation-only survey polled 519 software developers and 167 business managers from companies such as Fujitsu, NEC and Oracle to gauge trends in open-source and Linux development.

The result was the **Linux Foundation's 2014 Collaborative Development Trends Report**, which shows that companies view collaborative development as a key to future success. The report projected growth and investment in collaborative development to continue because of its tangibly faster project time to market, improved business relationships and lower development costs.

The key findings of the report were:

• Ninety-one percent of business managers and executives said collaborative software development was somewhat to very important to their business. Close to 80% also said collaborative development practices have been seen as more strategic to their organization over the past three years.

• Investments in collaborative software development are on the rise. Forty-four percent of business managers and executives said they would increase their investments in collaborative software development in the next six months; 42% said they would sustain their current investment, and no one reported they would decrease their investment.

• Sixty-three percent of developers said they spend more time now on collaborative software development, compared with five years ago. Fifty-nine percent reported increased participation in collaborative software development in just the last year.

• Eighty-three percent of developers said they benefitted personally from collaborative development through exposure to new tools and practices. ▮

The Top 5 benefits

of collaborative development for software developers:

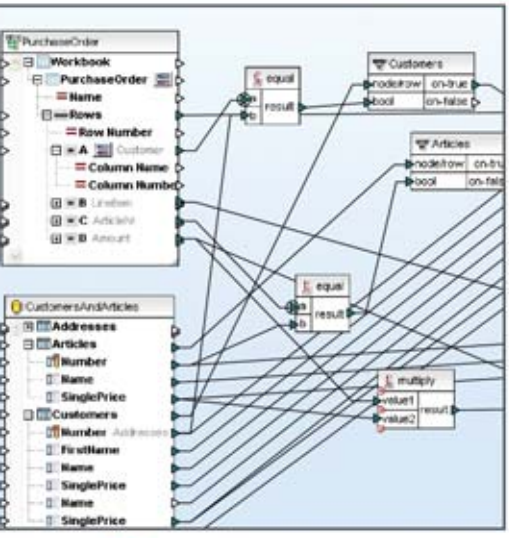
1. Exposure to new tools and development practices
2. Significant growth in skill sets
3. Large, strong knowledge base from which to draw expertise and support
4. Significant growth in professional networks
5. A sense of satisfaction in being involved in something bigger than any one company

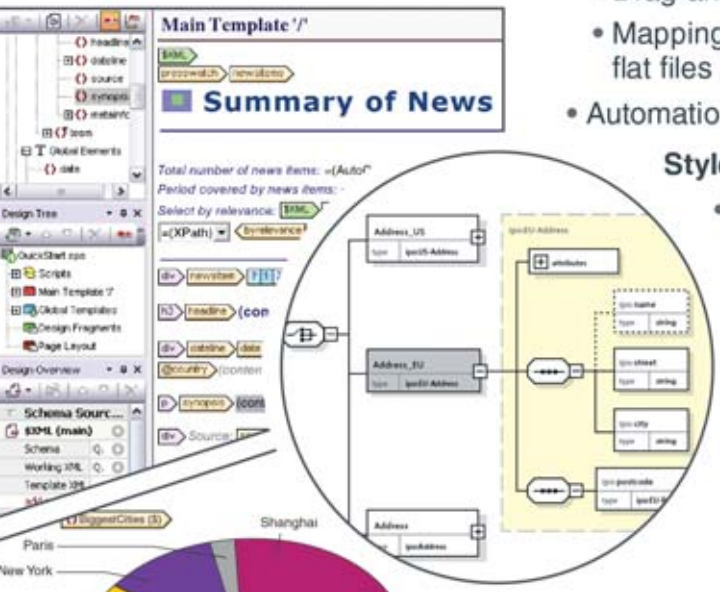
Bring your XML development projects to light with the complete set of tools from Altova®

ALTOVA® MissionKit®

Experience how Altova MissionKit®, a software development suite of industrial-strength XML, SQL, and data integration tools, can simplify even the most advanced XML development projects.

POWERED BY RAPTORXML®





Altova MissionKit includes multiple, tightly-integrated XML tools:

XMLSpy® – industry-leading XML editor

- **NEW!** Support for XML Schema 1.1 and XPath/XSLT/XQuery 3.0
- Industry's strongest validation engine with Smart Fix
- **NEW!** Powered by RaptorXML® for lightning-fast validation & processing
- Graphical editing views, powerful debuggers, code generation, & more

MapForce® – any-to-any data mapping & integration tool

- Drag-and-drop data conversion
- Mapping of XML, DBs, EDI, Excel®, XBRL, flat files & Web services
- Automation via MapForce Server

StyleVision® – visual XSLT stylesheet & report designer

- Graphical XSLT stylesheet & report design for XML, XBRL, & SQL databases
- **NEW!** XPath 3.0 and XSLT 3.0 support Powered by RaptorXML
- Output to HTML, PDF, Word & more
- Automation via StyleVision Server

Download a 30 day free trial!

Try before you buy with a free, fully functional trial from **www.altova.com**

ASPOSE.TOTAL

Powerful APIs which enable developers to harness the complexity of file format processing within their apps.

Your File Format APIs

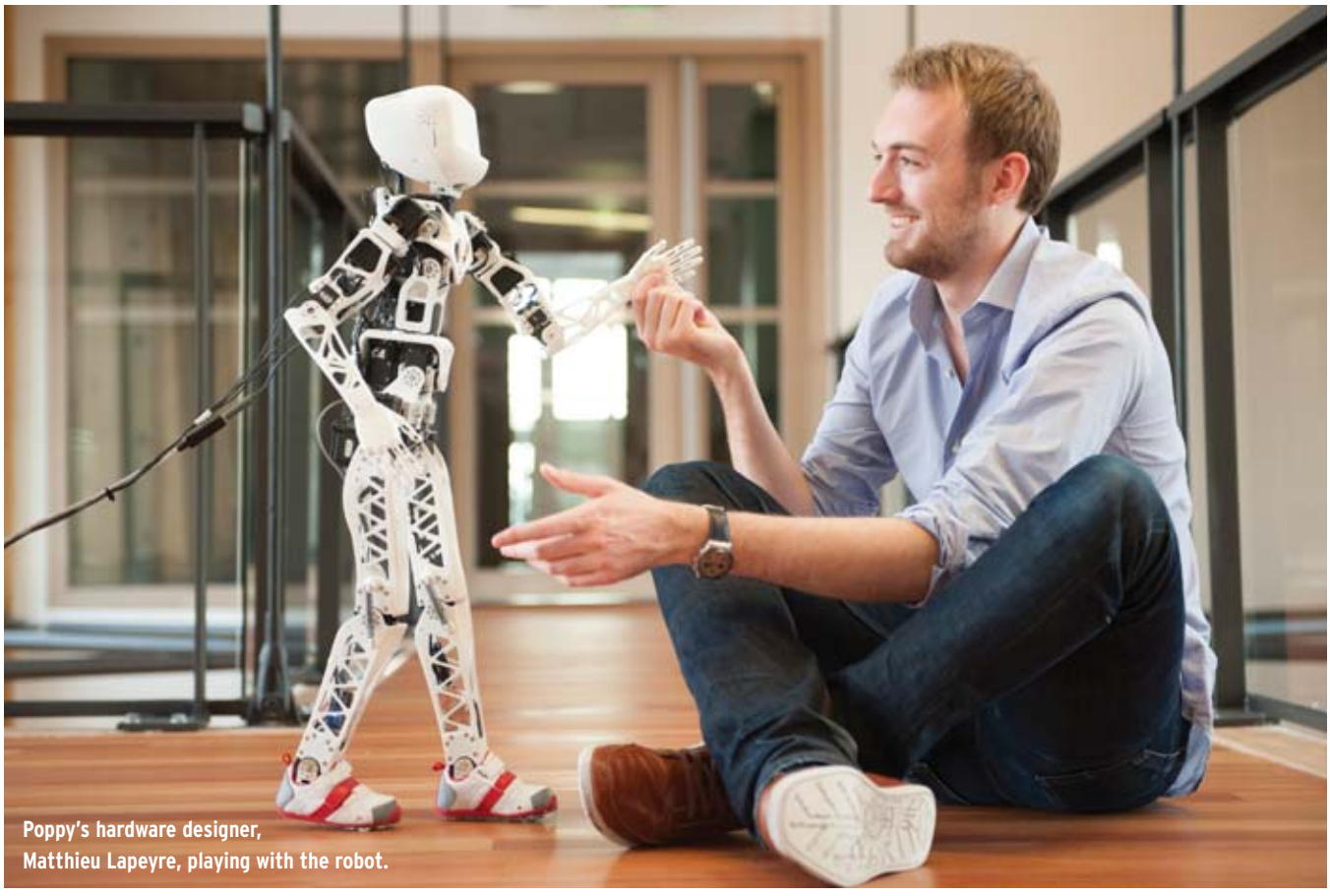
- **Aspose.Words**
  DOC, DOCX, RTF, HTML, PDF, XPS & other document formats.
- **Aspose.Pdf**
  PDF, XML, XLS-FO, HTML, BMP, JPG, PNG & other image formats.
- **Aspose.Cells**
  XLS, XLSX, XLSM, XLTX, CSV, SpreadsheetML & image formats.
- **Aspose.Slides**
  PPT, PPTX, POT, POTX, XPS, HTML, PNG, PDF & other formats.
- **Aspose.Email**
  MSG, EML, PST, EMLX & other formats.
- **Aspose.BarCode**
  JPG, PNG, BMP, GIF, TIFF, WMF, ICON & other image formats.
- **Aspose.Imaging**
  PDF, BMP, JPG, GIF, TIFF, PNG, PSD & other image formats.
- **Aspose.Tasks**
  XML, MPP, SVG, PDF, TIFF, PNG, CSV, MPT & other formats.
- **Aspose.Diagram**
  VSD, VSDX, VSS, VST, VSX & other formats.

... and more!

Aspose.Total for .NET
Aspose.Total for Java
Aspose.Total for Cloud
Aspose.Total for Android

Get your FREE evaluation copy at www.aspose.com

.NET Java Cloud Android



Poppy's hardware designer, Matthieu Lapeyre, playing with the robot.

THE PEOPLE'S ROBOT

Open-source Poppy Project aims for programmable automatons

BY ROB MARVIN

The robots are coming. Advances in robotic production and artificial intelligence are moving humanity progressively closer to the sentient beings of science fiction lore. And Google has bought 12 emerging robotics companies in the past year, secreting the technology away to experiment with in a lab somewhere in the depths of Googleplex.

Google's quest to create a real-life Android may sound a bit too close to the dystopian beginnings of "The Matrix," "The Terminator" or "I, Robot" for comfort, but on the other end of the spectrum is an open-source robotics movement aiming to bring cheap, accessible robotic software and hardware to the world.

The Poppy Project is an open-source humanoid platform with child-sized proportions, designed by the Bordeaux, France-based Flowers Laboratory, in conjunction with the French Institute for Research in Computer Science and Automation and ENSTA ParisTech. What began as a research project to study human development and motor skills evolved into Poppy: a humanoid platform consisting of an open-source Python library and framework along with 3D-printed modular hardware, designed to provide an affordable and hackable robot for science, education and art.

"Open source both for the skeleton and software is especially attractive because it's made so that people can hack it, transform it, improve it," said Pierre-Yves Oudeyer, research director and founder of Flowers Lab. "There is no entry barrier to get your hands on these technologies.

"We created the Poppy humanoid robot initially to study the properties of the legs, of the vertical column in balancing and dynamic walking, and then very quickly we also decided to make this robot open source. We wanted to share with other labs around the world in a community of science, so that not only could other labs reproduce what we're doing, but they can use the tools we've developed for their own research projects so they don't need to reinvent the wheel."

Prototype Poppy

The Poppy Project began three years ago as a product of the 3D printing boom. Oudeyer explained that traditional manufacturing techniques made the creation and, more importantly, the continued innovation of a humanoid robot prohibitively costly and time-intensive, but the arrival of 3D printing allowed the members of the FLOWERS (FLOWing Epi-

genetic Robots and Systems) team to print, assemble and improve their robot design at a rapid pace.

Poppy's hardware, designed primarily by FLOWERS team member and Ph.D. student Matthieu Lapeyre, allows for assembly in two to three days, with the sum of its parts costing about US$10,000.

Poppy is built for rich physical interaction. Behind the mechanisms controlling Poppy's leg, hand, arm and torso movements, wide-angle cameras and LCD screen is a cross-platform library written in Python called PyPot.

The PyPot framework, hosted on GitHub, works on Linux, Mac OS and Windows to enable simple development, deployment and scripting governing custom-built robots. PyPot defines low-level motor commands as well as higher-level primitive behaviors such as its gait. The framework then combines primitives and commands to create complex behaviors.

"The main idea was to build something that was as easy as can be, something really, really simple," said Pierre Rouanet, a FLOWERS team research engineer and the programmer behind PyPot. "The robot is not meant to be used solely be computer scientists, but other people who are not as expert developers."

Rouanet continues to work on PyPot, improving Poppy's control over balance, walking and other behaviors, but he hopes other developers will augment and reimagine what the open-source software is capable of.

"I would like to see the software be forked and used in many, many different directions," he said. "As we've tried to make it very generic, it could be really nice for people to use it to write different behavior for the robot. That behavior could be combined and reused by other people, by other labs. I'm sure they could do many thinks we haven't even thought about."

Building with Lego bricks

Poppy is a platform for experimentation, not only in terms of computer science, but also as part of a growing community of programmers, institutions and artists.

One space in which Poppy has taken hold is education. A number of engineering schools and universities have reached out about using Poppy as a teaching tool to show the integration of mechanical engineering, computer science and electronics in a single machine. The Flowers Lab is also working to make Poppy programmable with visual programming languages and environments such as Scratch, to teach coding to students.

"The open-source philosophy is attractive to educators, because if this is about learning how Poppy is walking, you really need to be able to open it, to deconstruct it, to see how it's built and how it's been designed," Oudeyer said.

Artists have also embraced Poppy, and the humanoid robot recently finished up an artistic residence at the Lycée Sainte-Famille chapel in Bordeaux. A dancer, a musician and a plastician were engaged in the first artistic application of Poppy as an expressive tool, to explore the theme of motion.

"The Poppy platform is interesting for [the artists] because of its association with movement," Oudeyer explained. "They set up a show where the humanoid robot is dancing with the dancer, and they're exploring how the coupling of the movement between the human and the robot can be done in such a way that it's producing emotional feelings."

The FLOWERS team is currently working on simplifying the open-source software and hardware design of Poppy even further, with a major "1.0" release set for September, according to Oudeyer. Beyond that, the team is focusing on fostering the community to advance the technology beyond their laboratory. The researchers see Poppy as a set of Lego bricks that could be used to build a new and diverse generation of widespread, accessible robots.

"It's really a platform that is providing Lego bricks for building animated structures, that would teach people how to use these technologies and then allow them to advance innovative products," Oudeyer said. "Poppy is one humanoid, but from this platform many groups of people could invent variations such that this humanoid creature could be transformed into families and various subfamilies." ▮

Jasper expands voice controls

Open-source components power platform

BY ROB MARVIN

Two Princeton University undergrads have developed an open-source platform for building always-on, voice-controlled applications for any device.

Created by Charlie Marsh and Shubhro Saha, both Princeton University undergraduates, Jasper provides applications with voice control that is always active. As with other voice-control software, such as Apple's Siri and Microsoft's newly announced Cortana, users control a Jasper-enabled application by speaking its name, followed by a command.

Applications integrated with Jasper can perform Internet searches, update social media platforms, notify the user of e-mails, or control a music player. The Developer API is configurable in a standard (user initiates contact with Jasper) or notification (Jasper notifies the user of something) mode to create custom controls for an application on any given device. The platform is integrated with services such as Facebook, Gmail and Spotify, the last of which Marsh and Saha wrote a specific set of commands for.

"[Jasper] has a dead-simple API," said Marsh in a promotional video. "As a developer, you can build your own voice-enabled applications to control anything."

Jasper was created using a collection of open-source libraries. It uses PocketSphinx to perform speech recognition; eSpeak to generate its voice; and Phonetisaurus for on-the-fly creation of dictionary and vocabulary models to conform to the user's speech.

The platform's hardware consists of a Raspberry Pi Model B computer along with a generic microphone and network adapter, and Jasper runs on the Debian-based Raspbian operating system.

Jasper's APIs and documentation are available on GitHub.

Single-page apps: The new normal

JavaScript advances a novel way of making programs

BY CHRISTINA MULLIGAN

Single-page applications (SPAs) are finally starting to get some recognition, according to Dru Henke, executive director at NVISIA, who predicted in the next few years that all applications developed will be SPAs.

SPAs have been around since the early 2000s, but due to the lack of expertise and tools, they haven't really had much popularity.

"They really kind of started in their emphasis about 10 years ago, but were difficult to write at that point in time," said Henke. "Very few people had the expertise and saw it as too big of a challenge to take on in the mainstream enterprise setting."

Today, with the variety of tools and IDEs, as well as a growing level of JavaScript expertise, creating SPAs is becoming easier, Henke explained. JetBrains offers IntelliJ IDEA. Google offers AngularJS and Google Web Toolkit. Ember offers Ember.js. And there are many more.

How are single-page apps different from applications that separate the presentational layer from the business layer? They actually aren't all that different. Single-page applications, unlike traditional applications, are apps where instead of a page refreshing every time a user clicks on a link, only small bits of the application refresh. From a user perspective it seems that they are staying on the same page the whole time—just like separating the presentational layer from the business logic. These types of applications are becoming more and more popular due to thetools and IDEs that have evolved over the past couple of years.

"The tools have matured and the frameworks are now more sophisticated, so developers don't have to write so much plumbing from scratch," Henke said.

"Users never really leave the page," said Andrew Connell, a developer who specializes in Microsoft SharePoint and content-management systems. "Rather, by leveraging JavaScript, CSS and HTML fragments, parts of the page are changed depending on the actions performed by the user. It gives the user more of an 'application' feel rather than a Web page feel, without having to install anything."

SPAs are different from loosely coupled architecture, but they fit effortlessly together, according to Henke. Loosely coupled architecture separates the business layer from the presentation layer using JSON and/or XML, while SPAs use JSON and/or XML in addition to HTML, JavaScript and CSS.

"Both of these data formats are very interoperable and 'neutral' with respect to server-side technologies and, generally, loosely coupled service layers can already speak in at least one of these formats," he said.

SPA benefits

One of the biggest benefits of SPAs is reach, according to Connell. "Where you would traditionally build a desktop or mobile client application, building an SPA allows developers to focus on one implementation that everyone can access, provided they are online and connected," he said.

According to Henke, SPAs in general are more sophisticated than traditional apps. "The challenge you get into when writing multipage apps is trying to maintain the state of what a user is doing," he said. "That state management becomes a limiting factor on how sophisticated your application can be."

In the enterprise, user experience is becoming more important. When call centers and other large-scale internal-facing apps went from being rich clients running locally to being Web-based, users gave up a lot of usability, according to Henke. Because of mobile, users have gotten used to sophisticated apps and expect that same kind of experience on their browser, and that is where SPAs come in play, he believed.

"I feel any enterprise that doesn't choose to go to single-page apps will find themselves on an island," he said.

Henke went on to explain that the level of efficiency in developing an SPA is a lot greater than just a page-based app where a user needs to load and reload pages after every action.

Important SPA tips

Making SPAs search-engine friendly is one of the most challenging things about them, according to Connell. "In the cases of an anonymous SPA, developers will need to ensure that not only they have a good deep-linking strategy implemented, but also a good way to expose this to the search-engine crawler," he said.

Developers also need to consider the online/offline status of an application. With SPAs, users can load the app and go offline, according to Connell.

"If you want to support this scenario, you need to make sure all network calls test for the connectivity state and inform the user when the state changes," he said. "In addition, you also need to typically implement some sort of client-side caching to hold onto changes until connectivity is restored to the app."

"If you are trying to present something that is slick-looking with a high degree of polish from a user experience level, then you are almost always going to be pushed to a single-page app because you don't have the screen refresh happening every time you do something in a browser," added Henke.

Lastly, when deciding whether or not to build an SPA, developers need to consider whom they are trying to target. "Some of the greatest improvements we've seen recently around the creation and development of SPAs depend on a more current browser," said Connell.

"Older browsers don't have some of the ECMAScript v5 property support that some of the popular SPA presentation frameworks use." ▮

Spreadsheets Made Easy.

Fastest Calculations

Evaluate complex Excel-based models and business rules with the fastest and most complete Excel-compatible calculation engine available.

Powerful Controls

Add powerful Excel-compatible viewing, editing, formatting, calculating, filtering, sorting, charting, printing and more to your WinForms, WPF and Silverlight applications.

Comprehensive Charting

Enable users to visualize data with comprehensive Excel-compatible charting which makes creating, modifying, rendering and interacting with complex charts easier than ever before.

Scalable Reporting

Easily create richly formatted Excel reports without Excel from any ASP.NET, Windows Forms, WPF or Silverlight application.

Download your free fully functional evaluation at SpreadsheetGear.com

SpreadsheetGear

Toll Free USA (888) 774-3273 | Phone (913) 390-4797 | sales@spreadsheetgear.com

How to get management on board with agile

Author Steve Denning talks about changing the focus from making money to delivering value

BY CHRISTINA MULLIGAN

Transitioning to agile often becomes an obstacle when teams have made the switch while management does not.

"The mistake that most organizations are making today is that the leadership isn't really participating in the transition," said Lowell Lindstrom, vice president of services at VersionOne, in a recent SD Times interview. "They'll sponsor the transition, fund the transition and support the transition, but they don't actually participate. They don't actually do agile. They don't actually practice the things that they are asking their team members to do."

Steve Denning, award-winning author and 2014 ALM Forum keynote speaker, said one of the problems is most management is focused on making money rather than delivering value to customers. SD Times had a chance to talk to Denning on transitioning management to agile.

SD Times: What is agile management?

Steve Denning: It involves focusing on delivering value to customers rather than the goal of making money. Although it in fact does make a lot of money, that's the result, not the goal.

The goal is based on self-organizing teams, rather than having individuals report to management. It is workers coordinated in these agile practices, in their various versions of Scrum, Kanban, lean and whatnot, but they basically set aside coordinating work by reports and plans and instead work in very short cycles, deliver value at the end of each cycle, and get feedback at the end of each cycle.

Those practices put together as a coherent self-reinforcing package have a hard time in an organization that is run on the old practices, and that's the drama being played out in many organizations.

Why is it important for management to be agile?

The management will say our objective is to meet out quarterly numbers, and so everything focuses on how can we meet our quarterly numbers, how can we make money. The whole focus of the work starts to be pulled away from delivering value from customers and making a quick return.

What the company starts doing is totally counter to the philosophy of agile, which is about delivering value to customers. So you have conflicts really at every turn because the management is looking at one set of values and measuring things with one set of measures, allocating resources and rewarding and punishing people based on those metrics. Then you have the agile people working on a totally different set of goals and metrics. You can try to set up buffers between the two worlds, but the track record of coexistence is not a happy one.

Why do you think management isn't on the same page when it comes to agile? Why are they having such a hard time?

Power in the market place shifted from the seller to the buyer and many organizations just didn't notice that paradigm had shifted, and so they keep on trying to do the same thing. They find themselves running harder and harder just to stay in place so as to not fall further behind, and they are extremely stressed. In a sense they know that something is wrong, but they haven't quite figured out what it is and they haven't by and large brought it into agile practices.

Do you think management is just unaware or unwilling to transition?

They are entrenched in some very bad habits, and in some cases they are hugely compensated for maintaining those habits, particularly the C-suite. It's very difficult to get someone to understand something when they are being paid not to understand it. That is the situation in many of these large organizations, but the economic forces are overwhelming and will drive these organizations out of business unless they change. The choice is change or die, and many may decide to die, but they don't have a choice to keep doing what they are doing.

How are agile teams and management supposed to work together, and what needs to be done in an organization to get there?

You have to introduce agile thinking throughout the whole organization.

Different goals, different ways of structuring work, different ways of coordinating work, different ways of values, different ways of communication, those are the core principles of agile. So an organization that wants to have the whole organization agile needs to run the organization on those principles. This is a big shift; I am not saying it is a simple thing to do. It is a huge transformation and it is a different way of looking at the world. It is a phase change. Going from ice to water, or it's a Copernican Revolution where the center of the universe is shifted. It used to be within the corporation, and now it's with the customers. It is a fundamentally different way of looking at and understanding the world.

How is an organization supposed to transition the entire organization to agile? Who leads the transition?

What you have in most big organizations are pockets of agile, even large, very large pockets. I mean in GE for instance, a very large organization that has a huge agile community in it, they are like revolutionaries within GE and actively agitating for the whole GE to become agile. You have GE largely still running on traditional lines, and then you have a whole segment of the corporation running on agile lines, so that certainly lays the foundation for change.

Ultimately, culture changes of this depth and magnitude require support at the top, and so it is great to have these islands of agile lower down, but ultimately the top of the organization has to come to terms with it and say, "We are going to run the whole organization in a different way." That hasn't happened in GE, but it will happen. It is only a matter of time.

I have heard companies say that in order for managers and agile teams to be on the same page, there has to be a level of trust. Do you agree with that?

Absolutely. Trust is key. That is a big problem in traditional management firms: There is very low trust. And that is one reason that it runs into problems. There is also verified trust when agile and Scrum are run properly. It is not blind trust. It has having a set of processes where there is continuous direct feedback from customers: Are they on the right track? If not, why not?

And in fact there is much more transparency than traditional management. It's that transparency of agile that often horrifies management because suddenly all of the tricks that traditional management play on are revealed, and so it becomes intolerable, and the management often backs off and says "We don't like the look of that," and goes back to their old ways.

Trust grows from actually listening to people and understanding what they are saying and then acting consistently with what you say, and if you start doing those things, then trust will build up. When you don't have that transparency and you don't have that consistency between saying and doing, then obviously trust breaks down, so those things are age-old factors in building up trust and destroying trust. You can destroy trust in about 10 seconds, and it takes a long time to build up trust. ▮

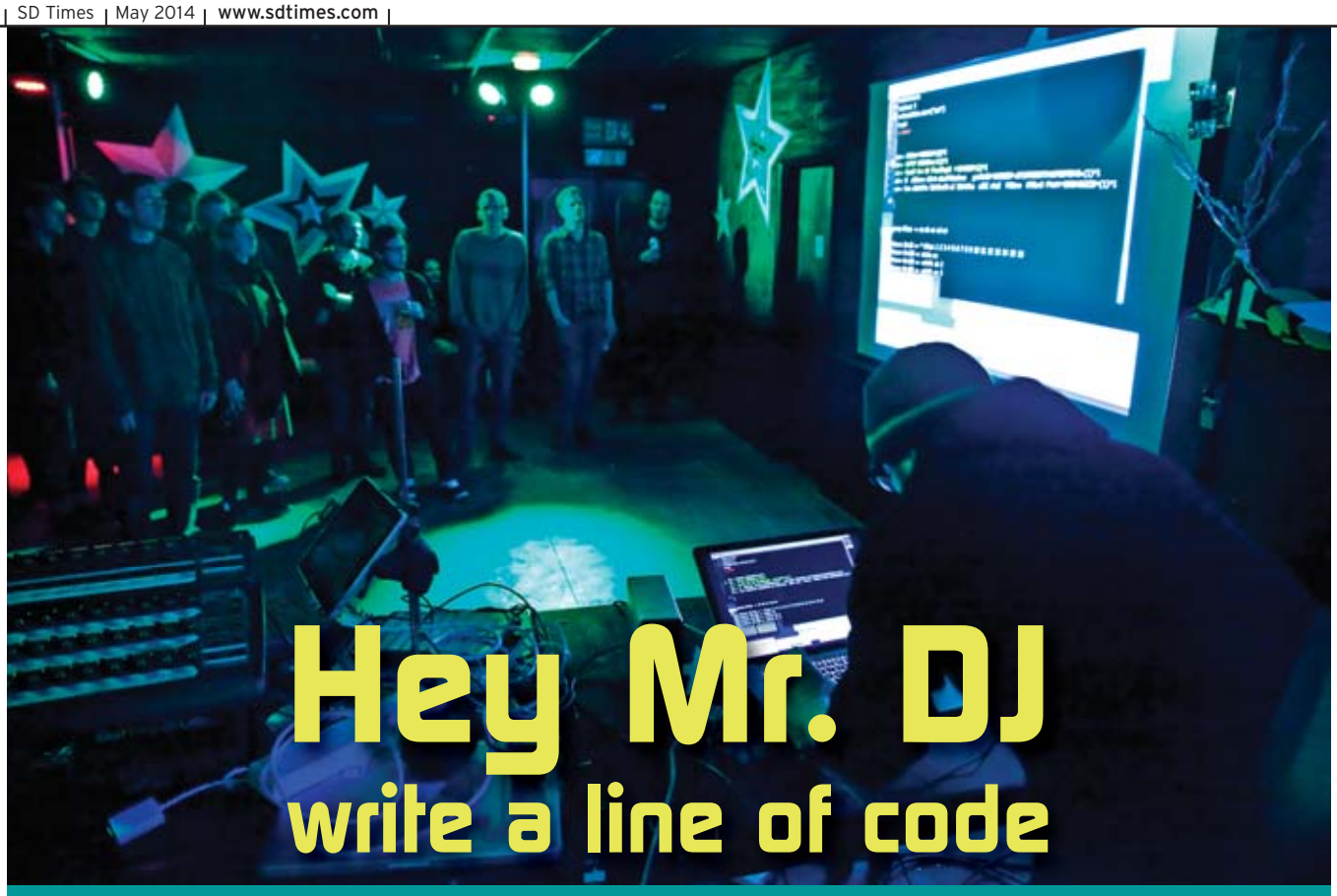


Photo by www.underyourskinphoto.co.uk

Hey Mr. DJ write a line of code

Algoraves, a growing trend where the music is programmed on stage

BY ROB MARVIN

A group of people stands in a darkened room staring at a blank screen. A line of code appears, and then another, and another. Disjointed electronic sounds emanate from speakers as a coder behind a laptop onstage builds the code methodically into a coherent rhythm, the code growing more layered and complex with each passing moment. The live coder's cursor moves frantically from line to line. The audience, their eyes transfixed on the screen, begins to dance.

What's happening is called an algorave, or at least the first few minutes of one. Algorithmic raves, an offshoot of a type of improvised algorithmic composition known as live coding, are a growing movement among both programmers and musicians. Over the past few years across western Europe and in places like Mexico, Australia, Japan and even as close as Canada, live-coded performances have given electronic artists and bands a way to break through the stand-around DJ culture of today's electronic music in live, messy fashion.

Live coding isn't supposed to be completely smooth. Performers are writing code on the fly, often in a programming environment built for live coding or using custom software they've created themselves. And by projecting their screen, audience members get to see the sound, and even become active participants, as the performance is created.

"Live coding comes from a reaction against laptop performance in music, where people won't be projecting their screen, they'll just be operating software to create music," said Dave Griffiths, a member of the live coding band Slub who also goes by the stage name NeboGeo. "Half the time they're playing MP3s and there's absolutely no engagement with the audience at all. It's very passive.

"The idea of live coding is to confront that, so when things go wrong, you get a very interesting effect with the audience. Everything goes silent, you see the performer panicking, their mouse is shooting around the code scrolling up and down like, 'What did I do?' Then everyone starts looking at the screen and trying to help. I've been in performances where people will scream out, 'You've missed a semicolon!' "

Slub has been live coding since 2000

Rooted in live audio

Algoraves are rooted in a 2004 live audio symposium, created by German philosopher and live coder Julian Rohrhuber, called Changing Grammars in Hamburg, Germany.

Early live coders and bands trekked to Hamburg, and what resulted several days later was something called the Terrestrial Organization for the Proliferation of Live Art Programming (TOPLAP), an organization to explore and promote live coding.

"It was recognized that there were people in a few different countries and on a few different platforms trying to live code, and what we meant by live coding was also up for debate," early live coder Nick Collins explained.

TOPLAP has served as evangelist and ambassador for the live coding movement for the past decade, connecting isolated pockets of live coders with the international community, and facilitating meet-ups, concerts, conferences and festivals. ▮

—Rob Marvin

with original members Alex McLean and Adrian Ward, and transitioned exclusively to live coding in 2005 after Griffiths joined the band. The commitment to live-coding performances was a way to shift the audience's focus onto the algorithms creating the art rather than the artists. Audience members seeing a single line of code evolve into complex sound can more easily relate what they're seeing with what they're hearing.

"It's such a ridiculous thing to do, to be in a nightclub, writing software with your screen projected," Griffiths said. "Actually one of the things I find fascinating is the response you get from other programmers is really like, 'What's the point? Why are you doing this? People shouldn't see this.' Whereas people who are not programmers, especially musicians or artistic people, are fascinated because they've never really seen the process of writing software and they don't understand that it's a creative thing."

Each member of Slub uses his own self-built coding environment. Griffiths, whose background is in computer graphics, uses a custom visual programming language that creates schemes of Lisp code. Instead of using parentheses, it uses custom structures to encapsulate different sections of the code.

Ward uses an IDE called Pure Events, a tracker-like JavaScript environment built in the style of an Excel spreadsheet. He writes a bit of code in each cell, which is timed to trigger in harmony with what Griffiths and McLean are coding.

McLean, who also performs solo under the moniker Yaxu, initially used feedback.pl, a self-editing Perl programming environment. More recently he created a mini-pattern manipulation language in Haskell called Tidal, which flashes the section of code being executed so the audience can see it more clearly.

"A lot of the inspiration comes from pushing technology and seeing how we can build things differently with this completely different emphasis," Griffiths said. "What I'm seeing increasingly which is really, really interesting is live coders who are musicians. They're not interested in programming in any other sense than for them as a performance, so they aren't interested in programming in itself, but they have very much adopted it as part of a musical experience."

Where it all began

On a road trip to Nottingham, England between live-coding gigs in late 2011, McLean and Nick Collins, an electronic musician and computer music researcher who performs under the pseudonym Sick Lincoln, invented the concept of an algorave. They tuned into a radio station playing happy hardcore (an upbeat genre of techno music) and decided they wanted to try programming some raves.

"Algoraves were created partly in order to change the emphasis of what people expect," Griffiths said. "If you put on a live-coding event, people really don't know what to expect. They're all standing around stroking their chins trying to work out what's going on. During an algorave, people come expecting to drink and have fun and dance, and that totally changes the emphasis, which I think is one of the reasons it's really taken hold."

Algoraves have been described as the meeting point

continued on page 28 ▸

Dancing to live coding

◂ continued from page 27

between hacker philosophy, geek culture and clubbing. Many of the performers are live coders, but all forms of algorithmic music are welcome, unlike the more narrow focus of a live coding concert.

The first algorave took place in March 2012 as a warm-up concert for the 2012 SuperCollider Symposium, an international gathering of SuperCollider users (one of the original and most popular IDEs for live coding and algorithmic composition). Among the performers were Slub and Sick Lincoln.

"I find it a bit bizarre actually these days, because in my mind I'm still somewhere in London in a converted public toilet that has been turned into a bar where there's only an audience of 25 people," Collins said. "There's one really dodgy projector, and you're trying to play on a really battered sound system. So actually having real music festivals, wanting to have proper club nights of live coding or algoraves, sometimes it's a little odd... The quirkiness of it keeps it vibrant as a subculture."

Collins, currently a professor of computer music at Durham University in the U.K., has done extensive research into live coding and algorithmic music on top of some Web and mobile app development of his own. He's also been active in developing and maintaining SuperCollider.

In November 2012, Collins and McLean travelled to Mexico City for the International Live Coding Symposium. There they were introduced to a vibrant live-coding community and one of its organizers: musician and live coder Alexandra Cárdenas.

"A year or so ago in Mexico City, we had the first international symposium of live coding, and we invited the guys," Cárdenas said. "We met them and it was so good for our community to get their opinions on live coding. It opened our horizons and we realized we could do many more things than we thought."

A live coder making waves in the U.S.

Live coding has spread to many countries around the world, but it hasn't quite taken hold in the U.S. Not yet, at least. There are scattered live coders across America, some in New York and California, and some in places you wouldn't expect.

Mike Hodnick is an independent software developer and live coder living in Minneapolis, where the midsized metropolitan area houses both a growing technology sector and a thriving arts and music scene. The live-coding community is only in its infancy, but if Hodnick has his way, the Twin Cities may well become a hotbed for algoraves.

"There's a healthy arts community here, whether that's galleries or experimental music, and we do have a very healthy programming community as well," Hodnick said. "So we've got the right circumstances, it just needs a little bit of nurturing."

Hodnick, a veteran C# and .NET programmer and JavaScript Web developer, had no idea what live coding was until this past Thanksgiving. A lifelong drummer and electronic musician, a friend turned him onto live coding and the algoraves going on in Europe, and he's been going full steam ever since.

"I've always had this kind of unique interest in combining sound with code," Hodnick said. "I don't look to create something with a catchy melody or a hook. I like to create things that are very dense, rhythmically. Whether that's layers of rhythms that interact with each other, or something that's extremely fast and dense and creating sounds that way." ▮

—Rob Marvin

Cárdenas, who is currently getting a master's degree in sound studies from the University of the Arts in Berlin, moved from Colombia to Mexico City in 2000 to be a composer and guitarist. Once there, she discovered the open-source software scene and began experimenting with live coding.

"When I perform as an improviser, I create my musical instrument here on the computer," said Cárdenas. "I imagine the sounds and I have the power and possibility to create and transform them as I want."

Cárdenas has performed live coding, in the U.K., Germany, Norway and Slovenia, and was one of the first live coders to start performing in Tokyo. She was also the keynote speaker at an international live-coding workshop in Chennai, India this past January, bringing it to yet another corner of the world.

At the behest of McLean, Cárdenas performed at her first algorave in April 2013 on the MS Stubnitz, an old German merchant ship docked at London's Canary Wharf. Other people have organized algoraves, but McLean's influence is palpable in all of them.

"I think people like the name 'algorave.' It puts some emphasis on having fun and not taking yourself too seriously," said McLean. "I also think non-coders are just interested to see code presented in a different way, as creative material, and not for any purpose apart from making some music for the present moment."

Their own paths to follow

No one takes the same path to live coding. Each musician or programmer who stumbles across it comes from a different background, a unique set of circumstances that led him or her there. That's part of the appeal; it means something different for each performer, and the nature of live coding ensures that each performance is different from the last.

Griffiths came from a computer graphics background. Collins started out as a piano player and composer before getting into computer music in college. Cárdenas studied classical guitar and composition in Colombia before moving to Mexico City.

"There's an element of live coding, this ambiguity about what's what," Collins said. "Sometimes it's tongue in cheek and sometimes it's deadly serious. Perhaps that's part musician humor and part programmer humor, and it's part of what I hope might be a bit of charm. There's something almost fatally stupid about attempting to go and program a computer live onstage. I can't claim that every live-coding gig I've ever been to has been packed and playing to mass crowds, but it's amazing how far it has come." ▮

COMPONENT WATCH

Text Control introduces new HTML5 rendering technology

New product changes how docs are edited in browsers

BY CHRISTINA MULLIGAN

Text Control wants to improve the way documents are edited in a browser. The software component company recently announced a new HTML5 editor to create cross-platform, cross-browser reporting templates on the Web.

"The exciting feature is the rendering technology itself in all HTML5 browsers, including Chrome, Firefox, Safari and Internet Explorer," said Björn Meyer, president of Text Control USA. "This is the first true WYSIWYG, HTML5-based Web editor and reporting template designer. Developers can give their users an MS Word-compatible editor to create powerful reporting templates anywhere, in any browser on any device."

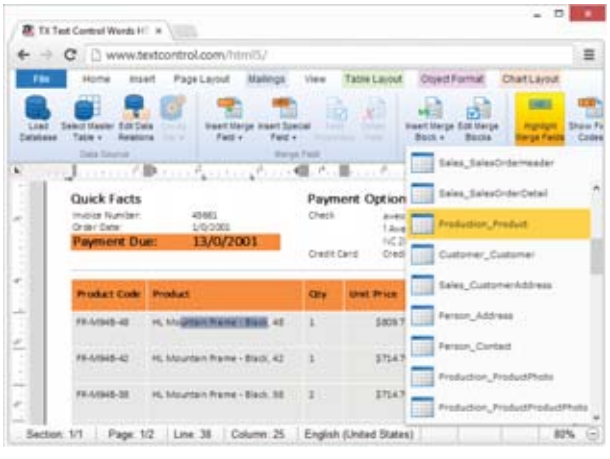
With the HTML5 editor, users can create and modify reporting templates; create PDF, DOCX or DOC files and view them in a browser; and merge templates on the server-side. The reporting engine of Text Control is used to merge templates in order to create documents and reports formats like PDF.

Users no longer have to use complex reporting designers, according to Meyer, because the HTML5 editor allows users to create templates with Word skills. Because of its Word compatibility, the editor can save and load industry-standard formats.

"Our new HTML5 rendering technology is a radical change in the way documents will be edited in thje browser," he said. "It won't be 'yet another HTML editor.'"

The rendering technology does not require client-side browser plug-ins; instead it uses pure HTML5 and JavaScript. In addition to HTML5 browser support, it also provides mobile device support for Android, iOS and Windows Phone 8.

Other features include support for all typical word-pressing features; reporting elements such as master-detail views; barcodes; and 2D and 3D charts that can be inserted and bound to database fields.



An example of the HTML5 editor in Chrome.

The HTML5 editor will be featured in the next version of TX Text Control Server for ASP.NET, version 11. "Text Control is going to provide a fully featured HTML5-based, MS Word-compatible editor with all word-processing features that are required," said Meyer. "It is a Web version of our reporting template designer to create templates in the Web." ▮

In other component news...

■ Windows content provider **Syncfusion** has released Essential Studio Enterprise Edition 2014, Volume 1. The release features 27 new JavaScript libraries designed for mobile development. All the libraries are compatible with the Knockout JavaScript library, and they provide Android, iOS and Windows themes. The release also features additional JavaScript libraries for HTML5-based Web development.

■ Document and content imaging solution provider **Accusoft** has announced updates to its Barcode Xpress software development kit. The SDK is used to add barcode-recognition capabilities to applications. Enhancements include improved handling of speckled barcodes; a new binarizer algorithm for superior accuracy when reading color and grayscale barcodes; improved detection of QR codes; and improved performance for reading PDF417 stacked linear barcodes. The SDK supports 32- and 64-bit .NET, Java, Java ME and ActiveX development environments. Accusoft has also released version 6 of its PDF Xpress SDK for enabling apps to create, modify and render standard PDF and PDF/A files. Features include making PDF files smaller with customizable settings to control the type and amount of compression applied, as well as a new automated function that analyzes PDFs in order to figure out ways to optimize image compression and reduce overall file size.

■ Microsoft component solution provider **ComponentOne** (a division of **GrapeCity**) has released an updated version of its suite of data and UI controls for Microsoft Visual Studio. Studio Enterprise 2014 version 1 features new theme support, built-in chart grouping and aggregation, and new chart types for financial applications. ▮

HOBBYIST LICENSE OFFER

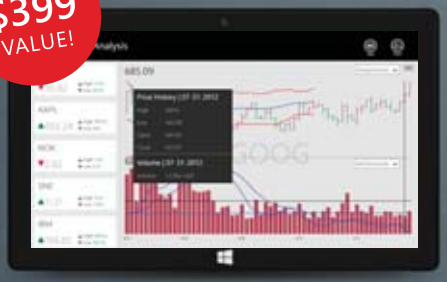
ONLY $1

$599 VALUE!

JavaScript

syncfusion.com/jshobbyistSDT

$399 VALUE!



WinRT

syncfusion.com/winrthobbyistSDT

$99 VALUE!

Windows Phone

syncfusion.com/winphonehobbyistSDT

- ✓ 30+ controls: charts, grids, gauges, and more
- ✓ One year of support and updates
- ✓ Individual developers qualify

Buy online and get your license today!

Visual Studio Partner

Syncfusion®

What kind of quality are you looking for?

Developers have to know the signs of defects in their code, but they can't do that until they know exactly what standard their code will be judged by

BY PATRICK HYNDS

Like beauty, quality is often in the eye of the beholder, and because perspective and values matter in how code is judged, it can be hard to nail down a proper definition for code quality with which everyone would agree.

For example, often a client will list code being "optimized" as a requirement. Sounds good, but what does that really mean? What should it mean?

For many, "optimized" means "speed." The code must be fast and performant. But that is not the only interpretation. In other situations, optimized code means that the code must

be easily maintained or portable to other platforms. In fact there are as many ways to interpret this word as there are dimensions to measure the quality of code. It all comes down to what is valued by those who will ultimately own the result.

This means that in many instances, the communications that describe the goal of any project are often more important than the technology ultimately chosen to implement it. This is especially true in the .NET development world because assumptions can affect where a project begins and what compromises are needed to get it done well enough to be accepted, or fast enough to hit a deliverable.

Code developed with any language on any framework against any platform requires this clear communication to have a chance to meet expectations, and there are other factors at work as well. So you have to ask yourself, what matters to you? What does optimized mean for your project? Think about that while we will cover the things that most agree lead to or define low-quality code.

The foundation matters

When setting out to construct anything, the foundation matters. Marketing, building construction and even literature succeed most when the opening or foundation is of high quality. Quality code has an even greater dependence in this regard since, in most cases, you are not writing machine language. We all depend on many abstraction layers, for better or worse, to do everything.

As far as foundations go, the .NET Framework is a good pick these days. It is mature and roundly tested for all manner of purposes. It is hard to imagine a use (suitable to a framework) that it has not addressed over the last decade.

continued on page 34 ▸

◂ continued from page 33

This argument for well-worn code being higher in quality is something component vendors emphasize as well, and it makes sense. There is a reason that adoption follows a curve where early adopters are followed at a distance by the masses.

At the BUILD conference in early April, Microsoft committed to open-source many libraries, including much of .NET itself. Never before have .NET developers had this much access to the guts of the platform, and there is hope that this act will increase the quality, security and stability of the .NET platform in the long run. But for now, .NET developers have to rely on the current quality of the platform. In most cases, this is a good dependency: .NET is well proven and has improved over several major iterations when we focus on the Base Class Libraries.

Other parts of .NET are less well worn and are even controversial for some. For example, Microsoft has never produced a data layer it did not ultimately replace. Even Entity Framework, the latest iteration of tools designed to make the data layer easy, has its detractors due to problems at scale and in certain high-volume situations. If the Entity Framework produces SQL under the covers that does not perform well in critical situations, that is definitely a knock on quality for the project. While not a concern for most, this serves as proof that the details matter.

The choice of Web service carrier is another area for .NET developers where there are choices, and the wrong choice can affect perceived quality of the project and the code. Would the goals of a project be better served by using Web API or SOAP via WCF? The answer depends on the details of the implementation and the requirements.

The bright spot in all of this is that, more often than not, the frameworks of .NET contribute positively to code quality because they handle so much of the grunt work that developers can concentrate on keeping their parts of the code up to snuff. Getting the architecture right raises the code quality, and these things raise more architectural questions to developers than ever before.

Code Smell examples

```
public void ContactStuff(int iCustID,
        string sName,
        string sAddr1,
        string sAddr2,
        string sCity,
        string sState,
        string sZip,
        string sPhone)
{
  // Encrypt it
  // RijndaelManaged symmetricKey = new
RijndaelManaged();
  // symmetricKey.Mode = CipherMode.CBC;

  // Bob was here
  switch (sCustType)
  {
    case "T":
      sType = "Trial";
      break;
    case "M":
      sType = "Member";
      break;
    case "R":
      sType = "Regular";
      break;
  }
  CRUD("INSERT",
     iCustID,
     sName, sAddr1 + " " + sAddr2,
     sCity + ", " + sState + " " + sZip,
     sPhone,
     sType
  )
}
```

These are the kinds of things you *don't* want to do.

Code smell

"It seemed like a good idea at the time..." is the post facto justification for many blunders, with common threads of not thinking about possible ramifications or just taking short cuts. Code smell represents the coding version of this often-heard lament.

Code smell is the term used by software craftsmen for things that seem like a good idea when a problem is not well understood or when essential use cases are not considered. Many of the cardinal sins of programming, including many deadly security mistakes, spring from this lack of foresight, but they can also come from rushing to get something to work in spite of understanding that the approach has issues.

This means that even experienced developers often produce code of low quality. In one such case, technical debt is incurred when expediency overrides doing things correctly. All software in the real world accumulates some technical debt as it matures. In some cases the gamble pays off if the product never matures to a place where these shortcuts need to be remediated, but in these cases you can be actually betting against your own success. The best approach to code smell is to avoid it and keep the books clear of technical debt, but if the debt needs to be incurred, then make a plan for going back later to do it right.

Code smells are also on the list of pet peeves for many developers. For example, a personal favorite has to do with not maintaining comments in code. It sounds like a trivial matter, and many of the code smells appear that way at first glance. But not maintaining comments often confuses the next developer who has to look at the code. Since developers naturally take comments at face value, it can take hours to fix code that behaves differently from what the comments seem to suggest. Not having comments is bad, but it's better than comments that lead to confusion.

Earlier I showed a sample procedure written in C# that presents a few code smells, including useless comments for code that no longer applies, old commented-out code, log parameter lists, and uncommunicative names. Each of these items can get your project in trouble by causing bugs, making the code harder to understand and generally slowing things down.

The uncommunicative names smell has to do with naming things such that someone who looks at the code won't know what should happen. In our example, the method name of "ContactStuff" only hints at what this proce-

continued on page 38 ▸

Need a helping hand developing your Document, Medical and Multimedia Applications?

Look no further than LEADTOOLS. It's comprehensive imaging SDKs include world-class support for ***OCR***, ***Forms Recognition***, ***Barcode***, ***PDF***, ***DICOM***, ***PACS***, ***Viewers*** and ***more***.

.NET Windows API WinRT Linux iOS OS X Android HTML5/JavaScript

DOWNLOAD OUR 60 DAY EVALUATION
WWW.LEADTOOLS.COM

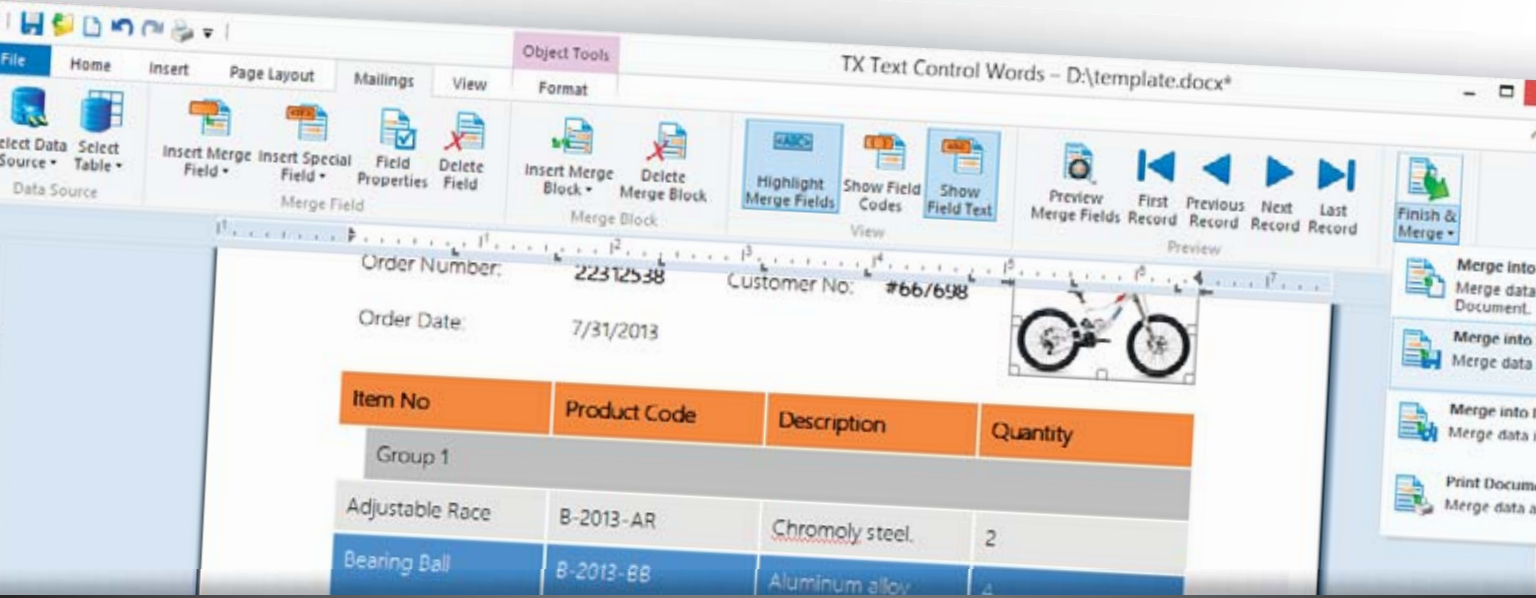
LEAD TECHNOLOGIES INCORPORATED

SALES@LEADTOOLS.COM
800.637.1835

TX TEXT CONTROL®
word processing components

Creating a report is as easy as writing a letter

- Reuse MS Word documents as your reporting templates
- Create encrypted and print-ready Adobe PDF and PDF/A
- Royalty-free WYSIWYG template designer
- Powerful and dynamic 2D/3D charting support
- Easy database connections and master-detail nested blocks
- 1D/2D barcode support including QRCode, IntelligentMail, EAN



www.textcontrol.com/reporting

txtextcontrol

US: +1 855-533-TEXT
EU: +49 421 427 067-10

Visual Studio Microsoft Partner

Reporting | Rich Text Editing | Spell Checking | Barcodes | PDF Reflow

Components and code quality

Building tools from scratch is riskier than buying from a vendor

BY PATRICK HYNDS

An experienced chef will say that the result can only be as good as the ingredients used; therefore, no discussion of .NET code quality would be complete without a discussion of the components provided by the most prominent vendors in the space.

Well-tested, flexible components can allow .NET developers to compose projects that have many thousands of lines of code of far greater quality than the developers would ever have time to bring together themselves. After all, a grid control that has been used in more than 10,000 projects has been debugged far better than a customized version put together by a corporate development team, even if that team has better programmers than the component vendor.

There are limits to this assumption, but field-testing and survival in real-world conditions do tend to bring many aspects of quality to light.

One potential danger is using a component that does too much and can introduce bugs, thanks to features that do not matter to the project at hand. In the end, buy vs. build will often favor getting a component from one of the vendors covered here.

ComponentOne

Greg Lutz, product manager for ComponentOne, talked about how keeping things simple contributes to code quality. While explaining the company's ClearStyle technology, he said, "We remove a lot of the complexity sometimes required to customize control templates to get the exact look and feel you want. ClearStyle includes supporting properties on the control that affect the appearance. Developers can very easily set up application-wide theming by just setting style properties rather than having to customize control templates, which can be a time-consuming and frustrating process."

The concept of simplicity as a vehicle to code quality came up again when Lutz explained that ComponentOne's XAML controls "support a unified namespace which helps make XAML markup more clear and concise."

DevExpress

Specializing in developer productivity, DevExpress products include CodeRush and TestCafe. They are perhaps best known for CodeRush, which makes it easier for developers to refactor their code and keep the drag of typing from getting in the way of producing quality code fast.

Mark Miller, chief scientist of IDE tools at DevExpress, talked about both of these products and how they contribute to the quality of code .NET developers produce. "High-quality code costs less to support, learn, maintain or change, and yields a higher degree of confidence (and customer satisfaction) when you ship as all test cases are green," he explained. He added that CodeRush has a debug visualizer and expression explorer that help .NET developers minimize bugs, and it can "create new test cases instantly, regardless of which .NET test framework is used."

TestCafe is also an interesting product for anyone doing Web development, though it is not specific to .NET developers since it does quite a bit for HTML5 projects as well.

PreEmptive

PreEmptive is known by most .NET developers because of its security product Dotfuscator, but a few years ago the company branched out with PreEmptive Analytics.

Described as a black box for your application, it delivers data about how things are working. Claiming to help "improve software quality, application adoption and user satisfaction," it helps developers deliver fast, efficient code of high quality. As mentioned previously, there are many ways to measure quality. Few or no bugs is a valid alternative measure to raw performance, and having telemetry that eliminates bugs is a good way to minimize bugs that cause crashes and slow performance.

However, if avoiding code smells is closer to the definition you seek, then while PreEmptive Analytics will help make things better, there is more needed in the quest for higher-quality code.

Telerik

Telerik considers code quality to be something its products must support across the software development life cycle. Stephen Forte, chief strategy officer for Telerik, pointed out that the company offers two products that target this role specifically during the development process. JustCode is Telerik's code productivity tool that "guides .NET developers through the refactoring process, often suggesting better code."

Refactoring tools are the key to ensuring that working code avoids code smell. The other end of the formula is the QA process, for which Telerik offers Test Studio. That software boasts rich Visual Studio integration. Forte asserts that Test Studio "stands out from the other .NET developer tools because it gives the developer the ability to collaborate with the QA team right inside of Visual Studio." ▮

The software craftsmanship movement

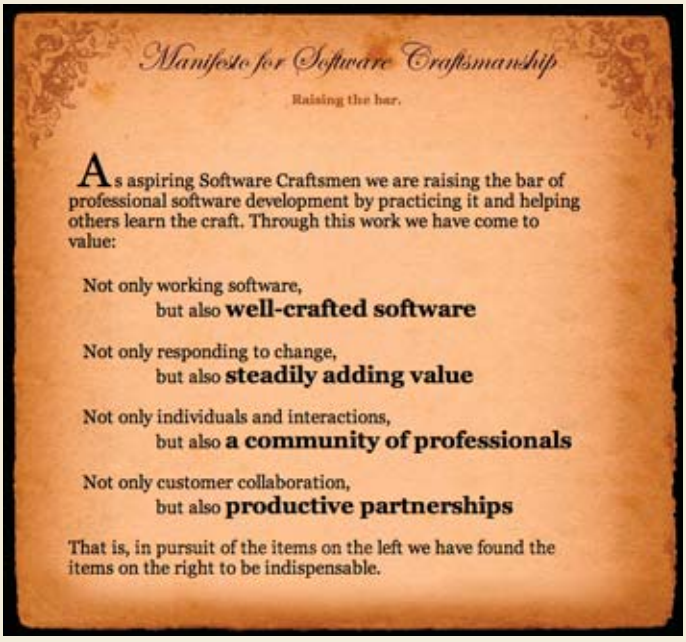
The first person who taught me about software craftsmanship is Steve Smith. He is an experienced trainer and development mentor who has published courses with Pluralsight and is very passionate about quality.

Software development as a profession has grown in ways unlike almost anything else in history. Only in the field of software development can someone with absolutely zero formal training command respect of those with advanced degrees in the subject and even rise to the top of the field.

One big reason for this is that for most of the last decade, there just have not been enough capable developers. However, the profession does lend itself to allowing people to prove themselves very quickly.

Getting the degree does not insulate developers from making bad judgments or being lazy, and with so many self-taught developers at work, the software craftsmanship movement provides guidance and resources to aspire toward continual improvements.

Bruce Backa, CEO of NTP Software, has managed a great many developers over the years. When asked why the software craftsmanship movement was needed, he explained it like this: "Most people are so happy when they figure out a solution to a problem that they immediately stop looking for alternative solutions and rush to implement." He went on to explain that, "in most cases there are a number of possible solutions to any given problem, but even assuming a small number such as four, this means that there is only a 25% chance that the first one that comes to mind is the best solution."



If you talk to any of the people like Smith who aspire to be a software craftsman, you realize that there is serious passion, but also an understanding that it cannot work if the devotees just browbeat everyone to not write junk code. After all, it is a movement driven by community rather than for profit. So it should come as no surprise that it has a manifesto to define the goals of software craftsmanship. The manifesto and its chief demands are reminders to not only do the job, but to also do it well.

People who study classical martial arts will know that a kata is a practice that you do that teaches you the movements for style of combat that makes up that martial art. The idea is that doing the kata will help you master the techniques and prepare you to apply them when needed in a confrontation. The word "kata" in Japanese means "form" and is an alien concept to most developers, though after understanding the intent, the average developer does warm to the idea.

SoftwareCraftsmanship.org hosts videos of code katas with straightforward tasks such as converting numbers to Roman numerals. The point of these exercises is to show correct approaches rather than to create code that will be used in an actual program. These would be great homework assignments for any aspiring developer who wants to produce quality code. (To hear more on Smith's views on software craftsmanship check out the .NET Rocks interview on the subject he did back in mid-November.)

—Patrick Hynds

◂ continued from page 34

dure is trying to accomplish. Any developer looking at it would have to look at the procedure itself to figure it out, which would slow things down.

Removing code smells is the best way to improve your code, all other things being equal. To learn more about how to do this, Smith has done a highly rated eight-hour course for Pluralsight named "Refactoring Fundamentals" that thoroughly covers how to address code smells and reduce technical debt by refactoring code properly. It is available with a subscription (or free trial) with Pluralsight. This is a great place to start for anyone looking to embrace the software craftsmanship movement and improve their code in the process.

Patterns and anti-patterns

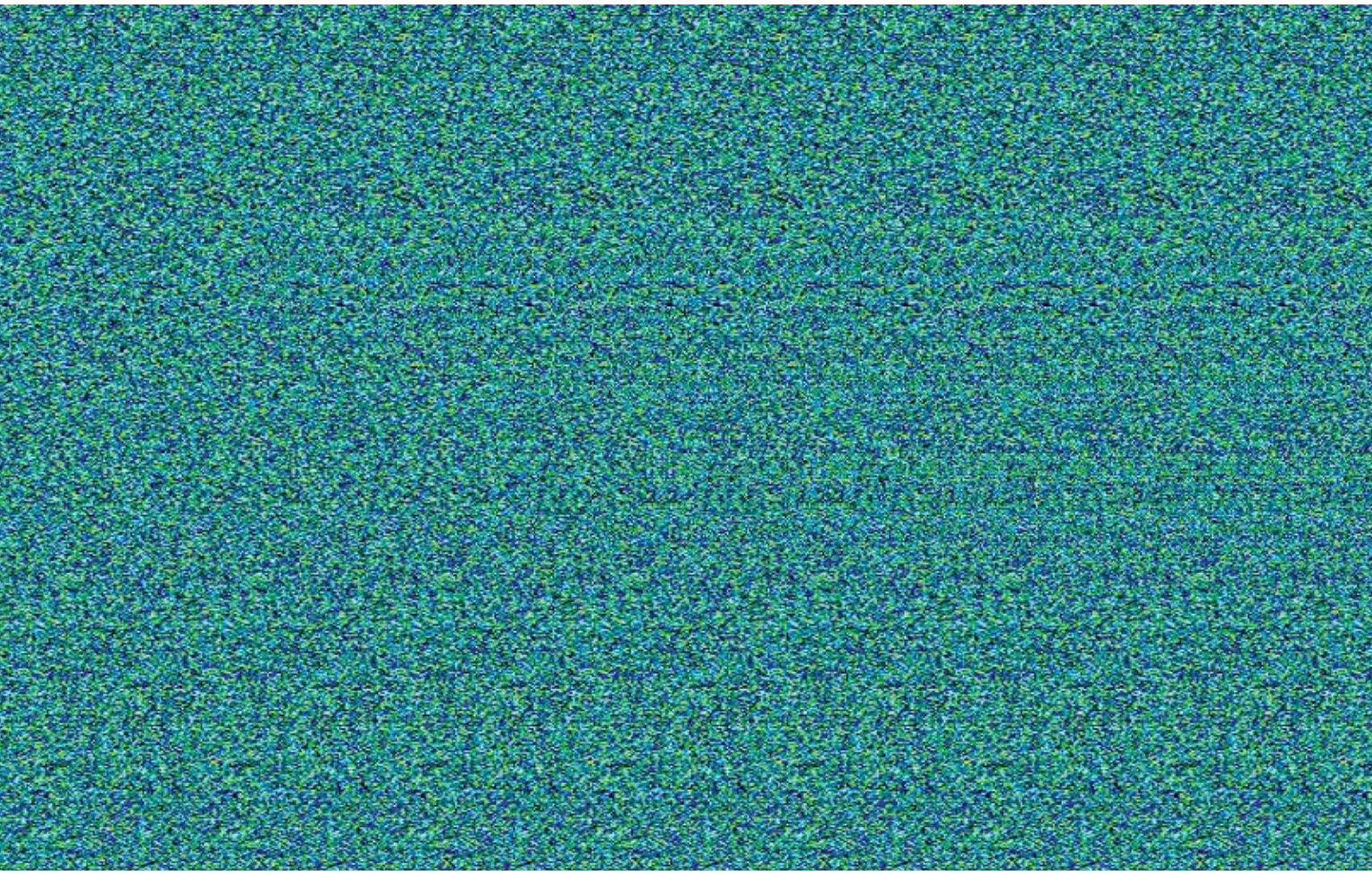
A design pattern shows you how you should be doing things and how components should be put together. And, by the same token, an anti-pattern talks about what you should not be doing and how things should not be put together.

Design patterns have been extremely helpful over the years, with Microsoft documenting them regularly via their Patterns and Practices group. The evolution of ASP.NET has been greatly influenced by the popularity of design patterns like ASP.NET MVC being based on the Model-View-Controller design pattern. These guides are meant to show developers how things are supposed to work, and they can illustrate things such as the best way to implement logging on a high-volume system without hurting performance, or how to deal with claims-based identity and access control.

Anti-patterns tend to be shorter than patterns and come across as rules, and can often be technology- or even language-dependent. For example, in the C# language, the Boolean data type

continued on page 40 ▸

The answer is right in front of you



Java Image Enabling SDKs that Help You See the Big Picture

Atalasoft JoltImage

At first glance it may seem difficult, but it's really quite simple. Atalasoft's JoltImage product is a proven SDK for image enabling your Java-based web applications, easily. Image enabling helps to add dimension to your data, so you can uncover insights such as correlations and causations hidden inside your 2-dimensional documents. Our SDK does the heavy lifting for you, saving time, money, and the headaches of figuring it out yourself. Backed by our highly knowledgeable & caffeinated support engineers, JoltImage will enable your success and make the big picture so much easier to see.

For tips on viewing this stereogram or to see the answer, visit go.atalasoft.com/magic

Image enabling experts & bacon connoisseurs. Visit us online to see our full line of SDK products for .NET and Java

Atalasoft
A Kofax Company

www.atalasoft.com

◂ continued from page 38

exists so Boolean comparisons can be done directly rather than comparing it to true or false.

Another anti-pattern is to avoid looking through a list of items in a loop when you can just ask for the item directly. The reason you might think you need the loop is to avoid trying to access something that does not actually exist (such as a file), but in those cases the logic goes that you can catch the exception if the file does not exist.

Often it will be hard to tell the difference between an anti-pattern and a code smell, and in fact the difference is somewhat semantic. Peter Ritchie addressed the issue in a blog post, saying that anti-patterns are "recognizable solutions (patterns) that don't work in at least one way and should never be used," and he goes on to point out that "code smells, on the other hand, are defined as '...a hint that something might be wrong, not a certainty.'"

Performance measures

Perhaps the hardest metric to satisfy for quality code lies in the area of performance. Users expect systems to be faster and faster to the point that instant is just good enough, and anything less is too slow. We now live in a world where the speed of light is actually getting in the way of stock market transactions, which is leading to trading companies competing to get their servers closer to the trading servers.

The laws of physics are literally the latest barrier in these instances. When developers set out to tackle a task that has performance as a component for measuring quality, the first critical step is determining the acceptable ranges. For example, if one is setting out to maintain a cache from remote servers, there are definite mechanisms to ensure maximum speed. But the responsiveness of the underlying network is usually an unalterable factor that must be taken into consideration.

Often developers and their project managers are brought to task to fix or justify their "slow code" when in fact the blame lies in the infrastructure. Companies like PreEmptive provide tools that help with this situation. Its Analytics product tracks everything your code is doing, and it can help locate where the delays are coming from so that slow code can be addressed and slow infrastructure can be called out when it is causing delays.

Too many projects are deemed failures because the results are too slow or not performant enough, when in fact the unspoken expectations for performance were never raised in a rational form at the start. It is important to understand that code quality often depends on prior agreement on what success looks like, and it can be completely separate from the code itself in certain areas. ▮

Read this story on sdtimes.com

Sanity helps

A popular video that made the rounds at the BUILD Conference called "The Expert" depicts a hapless engineer facing a room full of clueless stakeholders demanding impossible results. While exaggerated, these are precisely the kinds of things that must be stamped out early in a project to give the resulting code any chance of being acceptable.

Ultimately the best way to keep up code quality on a .NET project is to keep the requirements reasonable. Every other industry has realized this imperative, which is why we have laws that keep truckers from driving through the night on no sleep, and why there are studies required before medicines and surgical techniques are allowed. Winging it does not produce good, solid results, and code is no different.

Some of the rules that software craftsmanship seems to demand are less vital for an individual programmer who is not working with a team, but things change and assumptions are deadly, including assuming that no other developers will ever see the code being developed.

It is important to not confuse issues of code quality with controversies that are more akin to disagreements over religion. There are some issues that are hot-button items for many passionate people, with language choice and naming conventions being prominent examples. There are still some people who will disparage anything written in a language other than their chosen language.

More often than not it is Visual Basic that is getting heat from a small cadre of C# developers, though in recent years that has cooled down quite a bit. For naming conventions, there are some well-worn examples of what works and what does not, but the key is to focus on the code itself, since after it goes through the compiler, the names do not matter at all.

—Patrick Hynds

Cloud DevCon:
The Developer Conference for

Attend Cloud DevCon to get practical training in AWS technologies

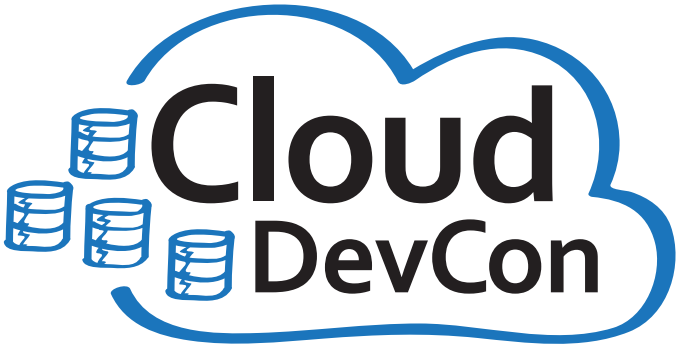
- Develop and deploy applications to Amazon's cloud
- Master AWS services such as Management Console, Elastic Beanstalk, OpsWorks, CloudFormation and more!
- Learn how to integrate technologies and languages to leverage the cost savings of cloud computing with the systems you already have
- Take your AWS knowledge to the next level – choose from **more than 55 tutorials and classes,** and put together your own custom program!
- Improve your own skills and your marketability as an AWS expert
- Discover HOW to better leverage AWS to help your organization today

Register Early and SAVE!

A BZ Media Event

CloudDevCon

Amazon Web Services



Check out this list of classes:

AWS Made Easy with Ansible and NetflixOSS
Optimizing Java Applications for AWS
Running a Big Data Stack in the Cloud: A Hands On Tutorial
Addressing Security Threats in the Cloud
Amazon Storage: Maximizing I/O for Optimal Application Performance
An Intro to Using Chef with AWS
AWS Automation and Deloyment with NetflixOSS and Ansible
AWS Benchmarking
AWS Development in .NET
Build it Worldwide: Hybrid AWS and Office 365
BURN IT DOWN *LIVE*: AWS High Availability
Cloudy With a Chance of Outage: Building for High Availability in the Cloud
Develop and Test Your On-Premise Applications in the Cloud
DynamoDB: A Study in Building Product on the AWS NoSQL Platform
How Overlay Networks can make AWS your Global WAN Network
Do More with Your Cloud Networks, not Hardware
Clustering in Public Clouds
Introduction to Elastic Beanstalk
Continuous Web and Mobile Quality in the Cloud
Customized Decision-Making Dashboards in the Cloud
Design Strategies for Building Scalable APIs on AWS
From a Physical to Virtual Infrastructure
How to Auto-Scale Applications on AWS Using Cloud Foundry as a Private PaaS
How to Do Cloud Multi-Tenancy Correctly in AWS
Ongoing Change: DevOps and the Brave New World
Introduction to the Big Data Ecosystem and a Hands-on Experience with Hadoop
Managing Development Environments from Laptop to Cloud with Vagrant
Missing Nothing: Application Insight in Hybrid Cloud Deployments
Performance at Cloud Scale: Seeing Is Believing
Practical Ways to Maximize Hadoop Performance on AWS
Running VMware Workloads on EC2 Without Modification

...and more!

June 23-25, 2014
San Francisco
Hyatt Regency Burlingame

www.CloudDevCon.net

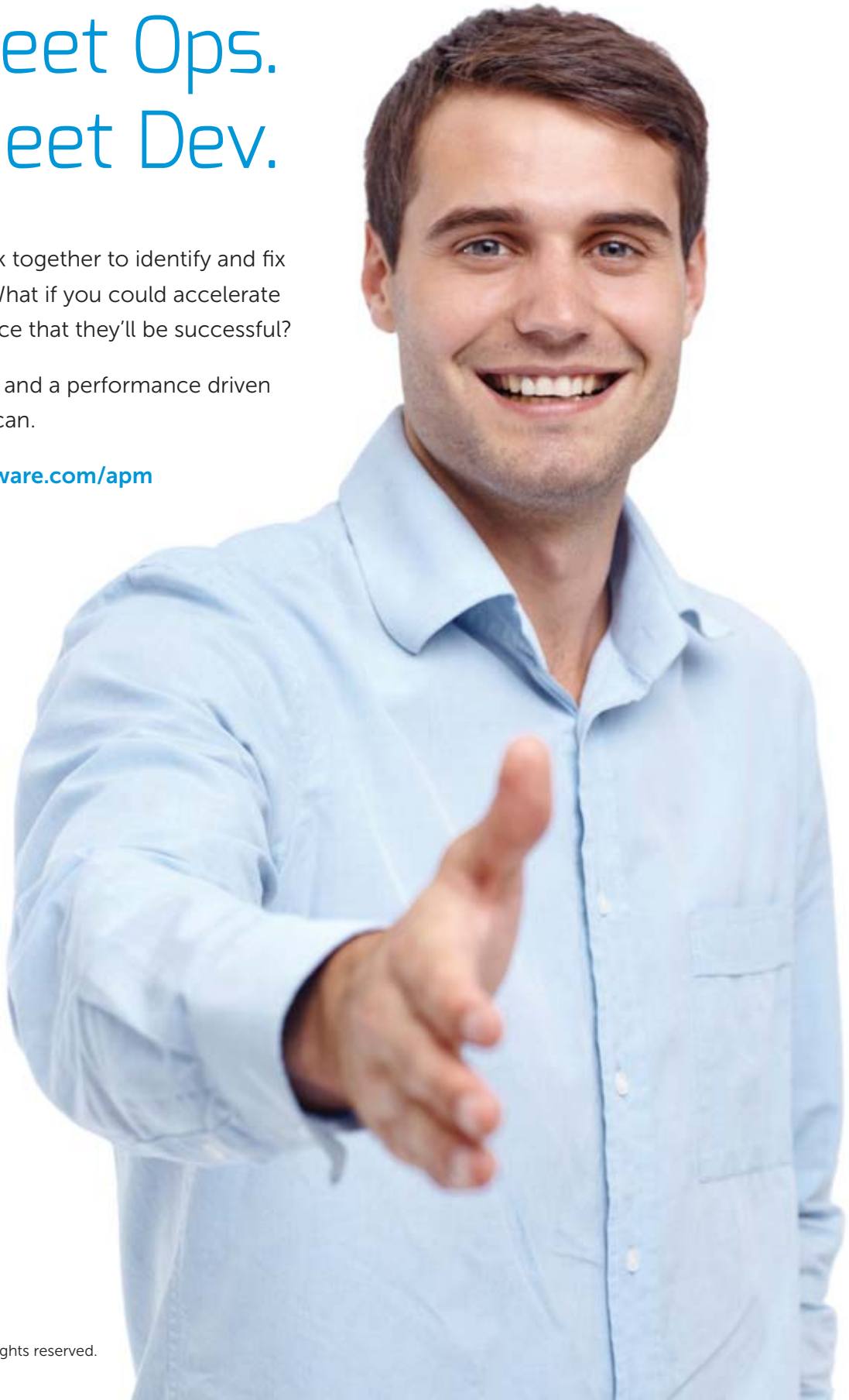
Compuware APM

Dev, meet Ops. Ops, meet Dev.

What if you could work together to identify and fix performance issues? What if you could accelerate releases with confidence that they'll be successful?

Through collaboration and a performance driven DevOps strategy, you can.

Learn more at **compuware.com/apm**



DevOps reality check

Planning, testing, mapping and more all go into a successful deployment, if you can handle it

BY ALEXANDRA WEBER MORALES

So you don't work at Facebook, Etsy or Netflix. Should you join the DevOps movement, now nearing the half-decade mark? With origins in the O'Reilly Velocity Conference, lean manufacturing, infrastructure as code and continuous delivery, DevOps, like agile before it, has captured the imaginations of practitioners, gurus, vendors—and now, executives.

"One of the things that has come out of the wash now that DevOps has moved from a bunch of enthusiastic people to a business problem is that a lot of people start asking, 'Why are we doing this?' DevOps is the goal, but it's not the goal, it's a means to achieve something," said Andrew Phillips, vice president of product management for XebiaLabs, a Boston-based deployment automation company. "Ironically enough, identifying that business goal is much harder."

In addition to defining the big, hairy business goals that faster delivery must drive toward, there's the question of whether infrastructure as code makes sense for legacy applications. "The continuous-delivery success stories are all in greenfield development," said Phillips. "The hidden premise is that if you can set things up right from scratch, you can do DevOps. But in a lot of companies, you have to make this work with an existing data center."

In a Velocity 2013 conference talk entitled "DevOps Isn't Just for WebOps," Michael Stahnke, software engineering director at Puppet Labs, said, "DevOps is a little weird. There's tons of press about it, tons of talks about it, and it's an echo chamber: The same people talking to the same people talking to the same people who agree with the same people...

"It didn't seem to apply to me. I was not ever doing 10 deploys a day. I did not have rockstar-ninja-pirates who were 'the best people ever.' I worked at a giant company. I was strongly discouraged from writing my own tools... We didn't have awesome developers; we didn't have any developers. And it was always in startups. I lived in flyover country."

Despite all those limitations, in his time at Caterpillar, he applied newfound ninja sys admin skills to remedying the database variance problem at the 85-year-old company and ultimately spearheading a DevOps transformation.

Does the DevOps diet really work?

Along with the increasing number of DevOps victories occurring outside of the cloud, several new surveys are looking at these practices to discover their popularity and how much of a difference they make.

Sponsored by Puppet Labs, Gene Kim and Jez Humble's 2013 State of DevOps Report had 4,039 respondents. Among its findings were that high-performing organizations deploy about 30x faster than low-performers, making changes in hours, not months. Outages, service impairments and other failures happened 50% less often for these high-achievers, who also boasted a 12x faster mean time to recovery.

Why does this matter? According to the survey, "High-performing organizations deploy at least once a week, and often multiple times a day. On average, this is 95% less time between deployments than lower-performing organizations, meaning they quickly respond to market changes or customer feedback, and iterate on new ideas."

Even when speed isn't critical, the complexity of deployments continues to rise. A 2013 survey by XebiaLabs found that nearly half of respondents faced a 20% to 30% increase in the volume of applications they managed. And awareness of DevOps is growing, with 68% of InformationWeek 2014 DevOps survey respondents professing awareness of the trend. Meanwhile, implementation is still for the leading-edge companies: Only 21% of InformationWeek respondents were "implementing DevOps."

The awareness and hype are well documented, then. So do the practices actually work? That's where Puppet Labs' survey gets interesting.

"We did find a couple things that were surprising that were not correlated with performance when we did the

continued on page 46 ▸

◄ continued from page 45

survey last year," said Humble.

The survey measured performance by asking:

- How often do you deploy code?
- What is the average lead time of a change (from "code committed" to "code successfully running in production")?
- What percentage of your changes require rollbacks or hotfixes?
- On average, how long does it take to restore service when something goes wrong?

Then, the survey asked about DevOps behaviors:

- Are environment and infrastructure changes checked into revision control?
- Is there an automated process to deploy environment and infrastructure changes?
- Who performs code deployments (Dev, Ops or both)?
- Who is on the hook for production support (Dev, Ops or both)?

"The first two behaviors were correlated with performance; the last two weren't," said Humble, meaning that developers doing deployments being paged when apps went down did not make a difference in continuous-delivery quality. He acknowledged, however, that team maturity could be one of several factors that weren't examined. Repeatability of the findings also remains to be seen: The survey was refined and taken again in January, and results will be available later this year.

Should developers carry pagers?

Starting with early DevOps proponent Patrick Debois, many DevOps proponents have talked about the importance of getting developers to empathize with operations. He defined "Devs wear pagers" as an intermediate DevOps practice for achieving this, writing, "By making [the] Dev feel the 'pain' of production problems, they will improve their code to cope with the problems."

Similarly, in a December 2013 QCon talk, Pedro Canahuati, head of infrastructure production engineering and site reliability at Facebook, declared that having his developers wear pagers was a

continued on page 48 ►

A guide to DevOps offerings

Amazon Web Services: OpsWorks provides an easy way for DevOps users to model and manage an entire application from load balancers to databases. With OpsWorks, users can scale their apps using automatic load-based or time-based scaling, and maintain its health by detecting failed instances and replacing them. It gives the user full control of deployments and automation of each component.

Atlassian: Bamboo is a continuous delivery server that automates build, test and deploy pipelines. When connected to Atlassian JIRA, Bamboo pushes build and test results to associated issues, and lets users track the code for each issue as it gets promoted to each environment. With automatic detection of new code lines, and the option to merge branches with each build, Bamboo is especially well suited for teams using branch-and-merge workflows like Gitflow or feature branching.

CA Technologies: CA offers **Release Automation** and **Service Virtualization** for DevOps. Release Automation is an enterprise-class, continuous-delivery solution that automates complex, multi-tier release deployments through orchestration and promotion of applications from development through production. Service Virtualization has the unique ability to eliminate constraints by virtualizing a target system's dynamic behavior, performance and data so the need for live systems is eliminated or reduced.

Chef: Enterprise Chef delivers a shared repository of code for automating applications and resources. The solution provides a way for development and operations teams to collaborate and move at the speed of the market. It includes role-based access control, centralized reporting, activity monitoring, an enhanced management console, and multi-tenancy.

Compuware: Compuware APM for DevOps focuses on three guiding principles: systems thinking, amplifying feedback loops, and a culture of continual experimentation and learning. The solution establishes performance as the lingua franca, has integration and collaboration across the life cycle built in, enhances agile development, automates performance testing, and accelerates release processes for continuous delivery.

IBM: IBM's DevOps solution bridges the divide among key stakeholders by addressing culture, process and tool integration across the software delivery life cycle, from ideation to delivery. The solution allows stakeholders to establish an essential enterprise capability for continuous software delivery that leverages lean and agile principles, removes waste, reduces time to customer feedback, and accelerates software delivery while balancing speed, quality and cost.

LeanKit: LeanKit provides a shared tool that's designed to allow development and operations to work collaboratively on a shared process. With LeanKit, teams can map their processes on virtual whiteboards to gain a complete and transparent look into status, issues and updates. The highly visual nature of LeanKit helps maximize effectiveness and keeps extended teams on the same page.

New Relic: New Relic provides real-time analytics DevOps teams need to deliver a stable environment in a rapid release environment. New Relic's SaaS solution monitors everything the application touches, from the browser and the server, to the code within application. Using New Relic's open platform, DevOps teams can monitor the entire application stack using simple-to-create plug-ins, giving them total freedom to collect and visualize the data needed to optimize the entire app environment.

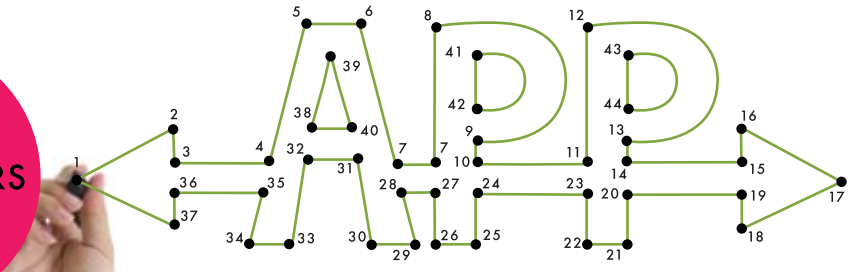
Puppet Labs: Puppet Enterprise is IT automation software that gives the power to easily automate repetitive tasks, quickly deploy critical applications, and proactively manage infrastructure, on premise or in the cloud. It

continued on page 48 ►

Do you have full DevOps visibility?

STACKIFY CONNECTS THE DOTS FOR YOU.

DEVELOPERS



OPERATIONS

stackify

Easily detect and resolve application problems and ensure your customers' satisfaction. Stackify combines monitoring, errors, metrics, logs and secure remote access with the relevant context in one platform.

Try it now for free. ▸ www.stackify.com/signup-sdtimes

Contact us for further info or demo at sales@stackify.com or 1-816-888-5055.

◂ continued from page 46

seminal practice that helped change the culture as they tackled technical debt and extreme operations scaling.

It may be too soon to tell if practices such as having developers on-call for production support or doing deployments themselves are critical or not. They are commonly cited DevOps hacks, nevertheless, on par with the sprint concept of agile development. Perhaps more critical, experts say, is the cultural change that should come first in any DevOps effort.

Culture shock: Rock stars need not apply

Coined in 2010 by John Willis and Damon Edwards, CAMS, or Culture/Automation/Management/Sharing, represents a stab at defining four critical aspects of DevOps. Jez Humble adds an L, for Lean, to the acronym. But all agree that culture is the first question to be addressed.

Starting with culture is critical in a world of vendor hype, said Ben Rockwood, director of cloud operations for Joyent. "A lot of people have heard of DevOps and all they hear is tools, tools, tools. 'DevOps is Puppet. DevOps is Chef.'" He counteracts that mentality with a focus on simplicity: "Everything is flow. Think about flow, flow, flow."

And don't worry if your team doesn't boast Joel Spolsky-approved peak performers: Puppet Labs' Stahnke points out that the spaghetti code he dealt with in operations at Caterpillar was written by none other than the founders of Puppet Labs.

"This is what I inherited: ISconf calling CFEngine calling ISconf calling CFEngine," he said in 2013. "So, I had rock stars. I had people who were the best in the business. They just didn't care. Or they weren't good at it. Or they were learning. Or they were experimenting. And actually, reading through some of that ISconf code, I can see where Puppet came from. Totally."

Start with mapping your value stream

If all this culture talk is getting a little confusing, step back a moment and take a big-picture view. "Do value-stream mapping and figure out where your bottlenecks are," said Humble, citing an oft-used lean manufacturing technique. "If you work on deployment automation, but your delay is not in deployment, you're not going to generate an improvement."

Though Debois calls a focus just on developers and operations staff "DevOps lite," author Gene Kim clarifies the reasoning behind this focus in his white paper, "The Top 11 Things You Need To Know About DevOps." "Why Development and IT Operations? Because that is typically the value stream that is between the business (where requirements are defined) and the customer (where value is delivered)," he wrote.

Lean manufacturing principles come from author Mary Poppendieck's application of the lessons she learned at 3M to software engineering. The first step for value stream mapping is to "put on your customer glasses," she advised in a 2011 talk. The process, in a nutshell, goes like this:

1. Define a process for mapping as the time from when a customer has a need to when that need is filled.
2. List key steps, the time each takes and the transition time between each one.
3. Calculate if any steps add value or are repeated.
4. Add up all the times for the total cycle time.
5. Divide the value added time by the total cycle time to calculate the process cycle efficiency.

Interestingly, Poppendieck points to a version 3.0 of software development thinking that may be the glimmerings of

continued on page 52 ▸

A guide to DevOps offerings

◂ continued from page 46

automates tasks at all stages of the IT infrastructure life cycle, including discovery, provisioning, OS and app configuration management, orchestration, and reporting. It includes event inspection, supported modules, role-based access control, certification management and VMware cloud provisioning.

- **Rackspace: DevOps Automation Service** is a managed support service for DevOps tools that helps organizations automate the process of deploying and scaling applications. The service saves time for developers and IT departments, and enables them to accelerate time to market for features. It also helps improve the quality of software deployments and to create more frequent software releases.

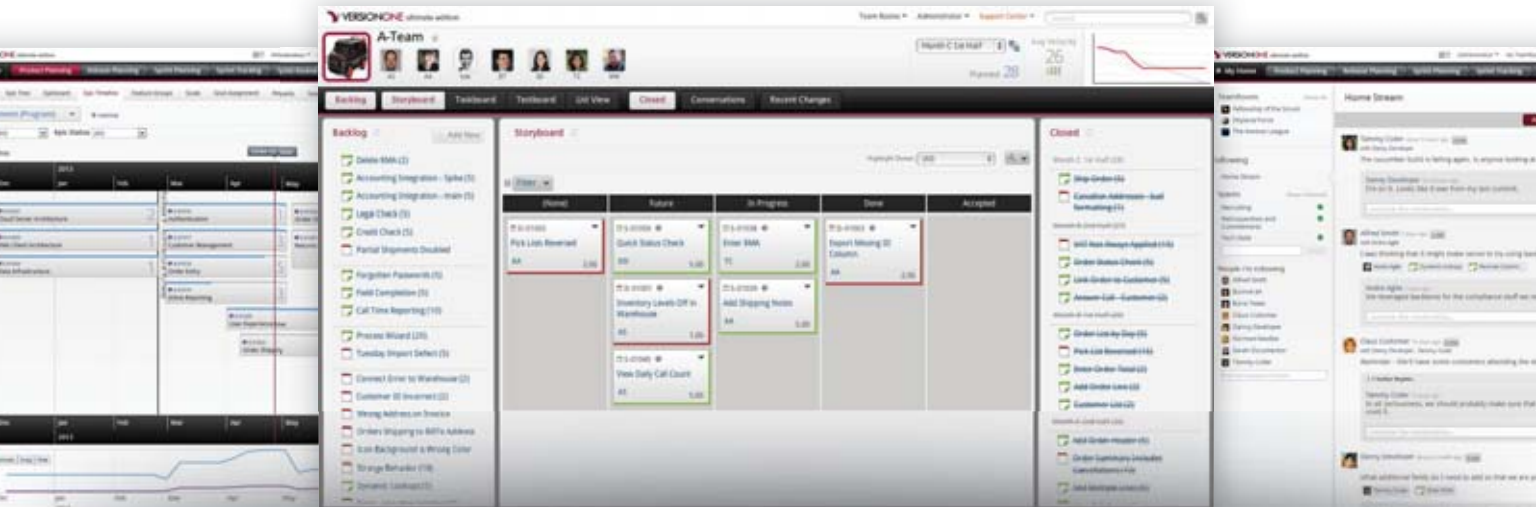
- **Ravello Systems: Ravello** is a nested virtualization provider offering a SaaS for DevOps, developers and IT to use the public cloud to develop and test on-premise applications. Ravello, powered by its Cloud Application Hypervisor, enables enterprise DevOps to encapsulate multi-tier applications and run them on-premise or in any cloud without making any changes. Features include the ability to replicate applications, create blueprints, and integrate with continuous-integration systems, facilitating collaboration among development teams.

- **Stackify: Stackify's solution** focuses on the application monitoring and troubleshooting part of DevOps. With its suite of tools, Stackify provides DevOps visibility and helps connect the dots between the development side and the operations side in order to get them to work together and ensure their application stack is working correctly. The platform combines monitoring, metrics, logs and secure remote access with the relevant context to monitor, diagnose and resolve application issues.

- **XebiaLabs:** With **XL Platform**, users can coordinate and automate all the steps required to get code from development to production: creating and scaling environments, deploying applications, managing and optimizing automated tests, and orchestrating production releases. The XL Platform helps teams adopt DevOps and continuous delivery in a step-by-step fashion, and transform an existing release process into an accelerated delivery pipeline while improving collaboration across Dev, Ops, QA and the business. ▮

VERSIONONE®

Agile Made Easier



Don't just take *our* word for it. Hear it from our customers:

An **intuitive** and **easy to use** UI gives this tool virtually no learning curve.

Intuitive interface made it **easy** to roll out across our organization.

Elegant tool that supports our process and doesn't get in our way.

Has taken our organization from **amateurs to professionals** in agile.

Easy to use interface enables users to quickly get the data they need.

Able to **quickly get the new teams up and running** using agile.

The #1 All-in-One Agile Management Tool for Teams of Any Size

See it. Try it. Love it.

www.versionone.com

Advice on succeeding with DevOps

These four experts warn against silver bullets, barriers and more

BY CHRISTINA MULLIGAN

SD Times caught up with some experts in the DevOps field to offer advice on how to be successful with DevOps and to debunk some of the myths people perceive about the process.

Stephen Franklin, CTO of LeanKit: The biggest misconception about DevOps is that it is IT operations with a smile, Franklin said.

"Many implementations try to break the traditional shared services model for IT operations by assigning a dedicated IT operations contact to a development team and instructing them not to grumble when the developers call. This misses the pivotal DevOps benefits of aligning goals, ownership, responsibility and measurement," he said.

In order to be successful in DevOps, Franklin said organizations need to be adaptive, open-minded and have a drive for results-based improvements. But there also needs to be a sense of teamwork in order for it to all come together.

"A successful DevOps organization recognizes that development and operations will have different goals, but efficiency comes when the two camps work toward a shared goal by using a shared process," he said.

Stephen Wilson, technical evangelist for Compuware: Wilson believes that thinking DevOps is all about automation is the No. 1 misconception.

"People tend to think, 'If I automate stuff, then I'm done,'" he said. "The problem is that if you automate and don't have a platform to understand the performance or manage the change near an environment, what you end up having is you are not building a pipeline of success and a delivery pipeline for features; you are delivering a sewer to move defects as fast as humanly possible from the developer's desktop into your production systems."

The key to being successful in DevOps is having a scientific curiosity, understanding how people work together, and pairing strengths with weaknesses, according to Wilson.

"You aren't necessarily going to build a team, but you are going to build a general skill set," he said. "You need people who want to take pride and quality in their work, and you need to find leadership that is good at fostering an environment where people can be honest with each other without blame."

"It isn't the technical expertise that you need in DevOps, but it is the team building and understanding that is going to springboard and push a DevOps culture forward in an IT organization."

Andrew Phillips, VP of product management for XebiaLabs: Phillips sees three common DevOps misconceptions. The first is that DevOps is a goal and not a means.

"DevOps is a fantastic initiative and set of ideas, but it is not a goal in itself," he said. "It is done in order to deliver value to your organization, but just because you are going to be DevOps doesn't mean you automatically are going to make your customers happier. Until you know what you are doing, DevOps is hard to measure if you are successful or not."

The second common misconception Phillips sees is that DevOps means installing a flavor-of-the month tool.

"Yes, tooling can help you address some common problems, but tooling is not the answer to DevOps," he said. "There is no silver bullet to DevOps."

Lastly, there is the misconception is that an organization can do DevOps by having a DevOps specialist team.

"DevOps is something we do across the organization as a team," said Phillips. "You can bring a specialist in to help you along with the process, but there doesn't need to be a separate team that tells teams what to do and what tools to use. DevOps is something we have to do ourselves. Everyone has to define for themselves what they are going to get out of DevOps."

'DevOps is a fantastic initiative and set of ideas, but it is not a goal in itself.'

—Andrew Phillips, XebiaLabs

Matt Watson, founder and CEO of Stackify: Watson stressed the importance of getting development teams and operations to work together.

"Naturally, operations and developers hate each other because developers make changes and rapidly push them out, the changes cause problems in software, and then operations is sort of on the hook to deal with those problems," he said.

The most important thing is avoiding finger-pointing and being knowledgeable. According to Watson, developers need to be more knowledgeable about some operations-related things, and operations need to be more educated on developer-related things. In order to get the two working together, he suggested having a common management that oversees both of them.

'Getting everyone on the same team and breaking down barriers is key.'

—Matt Watson, Stackify

"Getting everyone on the same team, trying to break down barriers and finding the different type of tools that can help with the visibility they need to deploy faster and measure success is key," said Watson. ▮

Read this story on sdtimes.com

YOU WRITE CODE.
WE GET IT TO YOUR USERS. SIMPLE.

START BUILDING YOUR CONTINUOUS DELIVERY PIPELINES WITH XL PLATFORM

- Start with your current release process and transform it into Continuous Delivery Pipelines.

- No More Deployment Scripts! Automate your deployments with a ton of out-of-the-box logic for on-prem middleware, cloud-hosted services, databases and more.

- Manage your Agile test suites: You have lots of automated tests running in many different test tools, and need to try to get a overall picture to analyze and optimize them? **We can help!**

- Create and scale your target environments: no more troubleshooting sessions trying to figure out why your app doesn't run in your test, QA or production environment.

- Analyze configuration drift: define, discover, track and compare application and system configuration from dev to production.

- Continuous Delivery Success!

BUILD ON WHAT YOU HAVE, ADD WHAT YOU NEED:

XL Platform integrates with Jenkins and all leading CI servers, as well as ALM, system provisioning and configuration tools, and much more.

THE SOONER YOU GET STARTED, THE FASTER YOU CAN GET GREAT FEATURES TO YOUR USERS. EVERYBODY WINS!

Start building with XL PLATFORM

XL DEPLOY
Agentless Application Release Automation

XL RELEASE
Enterprise Release Management For DevOps And Continuous Delivery

XL SCALE
On-demand Environment Provisioning

XL TEST
Complete Agile Test Management

Get Started Now!

www.xebialabs.com/xl-platform

XebiaLabs
Deliver Faster

◂ continued from page 48

a post-agile world: customer focus. Where version 1.0 was contract-focused and version 2.0 was development-focused, in 2010 she called the transition from "push" to "pull" the next software development inflection point.

Go ahead, automate something

Once the team has begun thinking lean and identified bottlenecks impeding flow, it's time to flip on the fun stuff: automation. The good news here is you don't have to be agile to automate.

"Some companies are thinking they first need an agile structure," said TJ Randall, director of sales engineering for XebiaLabs. "But there are a lot of areas you can improve on and automate. You don't have to be an agile shop."

What to automate? Here are some suggestions, courtesy of Kim and Humble's 2012 survey:

- Automate a single pain point such as DNS, NTP, or root passwords.
- Consolidate multiple sources of information into one source of truth by creating synchronization scripts for your HR system, CMDB, Asset DB, Policy DB, etc.
- Store all data inputs to configuration state in the configuration-management system "whether you use a service, a database (SQL or Hiera on disk), or pure data in version control (a YAML or JSON file)."

Kim's favorite DevOps pattern is to make deploying into the production environments part of the very first stages of development. "Ideally, the deployment mechanism we build is completely automated. Tools that can be used include shell scripts, Puppet, Chef, Solaris Jumpstart, Red Hat Kickstart, Debian Preseed, etc."

Make disaster fun

Anyone who's spent significant time in a cubicle dreads those interminable meetings with dull, risk-obsessed disaster recovery consultants. Blame Barack Obama for raising DevOps disaster recovery's profile even more.

A high-profile DevOps success was the 2012 election, in which President Obama's IT team made operational efficiency a strategic advantage. "'Game Days' were disaster preparedness exercises where DevOps simulated nightmare scenarios, such as a catastrophic database failure or Amazon's East Coast data center going offline," said Harper Reed, Obama for America's CTO, in published reports. "It's not enough to have it in a manual. The lesson of DevOps is that you actually have to practice and practice disaster recovery scenarios until you have them down cold."

Google's annual Disaster Recovery

Testing event is a multi-day exercise in finding systems and process vulnerabilities by intentionally causing failure. These tests start small and continually evolve, but often expose interesting cracks in the behemoth's armor.

"We simulated the earthquake by taking down a data center in the area that housed a number of our internal systems," wrote Google's Kripa Krishnan in the September 2012 issue of ACM Queue. "While the outage uncovered several services that were singly homed, it also exposed other interesting dependencies. For example, to avoid being affected by the outage, some teams decided to failover services from the data center to their workstations. Since the 'earthquake' occurred near Google headquarters in Mountain View (Calif.), the testing team disconnected the Mountain View campus as well—which meant all these failovers had failed."

A brain-eating zombie scenario helped Thomas Limoncelli discover that, while his Google team's N+2 redundancy meant it could survive two simultaneous data center outages, he felt a bug report was still in order: "We stayed within the SLA, but it was too close for comfort. Certainly we can do better," he wrote in the same ACM Queue article.

Look out for leprechauns

As software development evolves, a promising wrinkle is the attempt to apply the scientific method to evaluating engineering practices. In an effort to improve the statistical validity of his survey, Humble hired Nicole Forsgren Velasquez, a professor at the Jon M. Huntsman School of Business at Utah State University and an expert in survey design and analysis. Humble also pointed to a slew of recent books revealing the shaky foundations of many software development practices that turn out to be more belief- than evidence-based.

In "The Leprechauns of Software Engineering: How folklore turns into fact and what to do about it," author Laurent Bossavit punctures myths around the 10x variation in productivity among developers and other myths. "Making Software: What Really Works, and Why We Believe It" by Andy Oram, Greg Wilson and others advocates repeatable results, literature surveys, and rigorous qualitative analysis of software practices.

Be the curve

The good news is that embracing DevOps in 2014 puts your team on a path toward progress that many are just beginning to tread. Though case studies abound and new tools emerge daily, incremental improvement, architectural awareness and cultural sensitivity will go a long way, according to Humble.

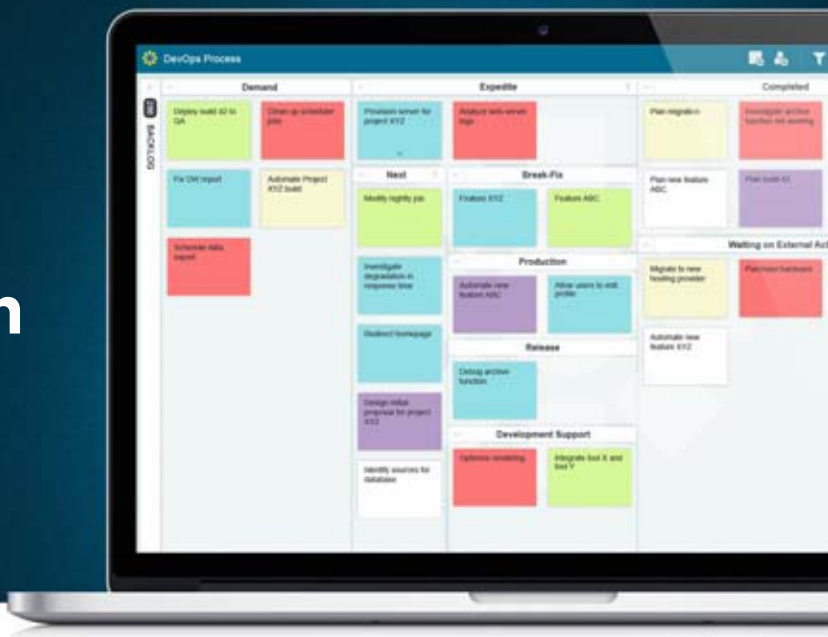
"DevOps is definitely becoming more widely adopted. It's no longer considered crazy talk, and you do see continuous delivery at scale," he said. So, in Biggest Loser fashion, is there a single success story he points to for new DevOps dieters? Not quite.

"On the one hand, I do see things like that; on the other, I urge caution," said Humble. "In complex systems, just copying the practices can't have the same results. Everyone has different situations." ▮

Read this story on sdtimes.com

LeanKit

Struggling with inefficient handoffs between Dev and Ops?



Increase communication and efficiency. Decrease confusion and error.

DevOps builds and deploys applications faster than ever before. But slamming together existing processes won't help you maintain speed and quality. You need a shared tool to keep your teams integrated and to visualize your work.

With LeanKit, you can map processes on virtual whiteboards for a complete and transparent look into status, issues and updates. Track and manage a multitude of tasks and steps. Keep extended teams on the same page. Clone boards to start your next release, and improve your process with automated reports.

Try it **FREE FOR 30 DAYS** *at* **LEANKIT.COM**

Integrations available for MS TFS, JIRA, GitHub, BuildMaster and other tools you use.

Available on the App Store

Get it on Google play

Responding to changes in Responsive Web Design

How this style of rendering pages according to devices got started, and where it may go next

BY ROB MARVIN

In its purest form, Responsive Web Design (RWD) is a client-side Web development technique that uses CSS, HTML and JavaScript to dynamically alter a website's front end across different screen sizes optimized for each device. It's one giant, all-purpose ball of code sent from the server that unpacks in the browser and renders itself to display on each device.

But how the technique is used and what the word "responsive" means have morphed as quickly as the device landscape around it.

RWD first caught on with Web and mobile developers as a solution for maintaining a single website across devices. Compared to building and managing separate mobile websites based on the different form factors, the idea of one codebase for one responsively designed site housed under one URL is an appealing one.

Though, as with all new technology, developers soon ran into RWD's limitations.

"From a perception standpoint, people are more mature than they were in 2010," said Forrester analyst Mark Grannan. "They greeted [RWD] as a silver bullet solution for all of your device needs. Everyone all of a sudden saw the tablet and the smartphone and didn't know what to do about delivering experiences optimized for those devices. So the idea of not having to create new teams and new systems was welcomed with open arms, but people didn't understand the full implica-

continued on page 56 ▸

◂ continued from page 54

tions of the complexity it would bring."

That complexity came in the form of performance slowdowns for transactional or data-heavy sites, a lack of flexibility in design across different devices, and the time-consuming development process to completely rewrite an existing website's code to be responsive.

To compensate, the term "responsive" has broadened over time to encompass differing techniques such as Adaptive Web Design, RESS (Responsive design with Server-Side components), and responsive delivery. These techniques incorporate client-side code, multiple device templates, and cloud-based code transformations that stretch far beyond the scope of RWD.

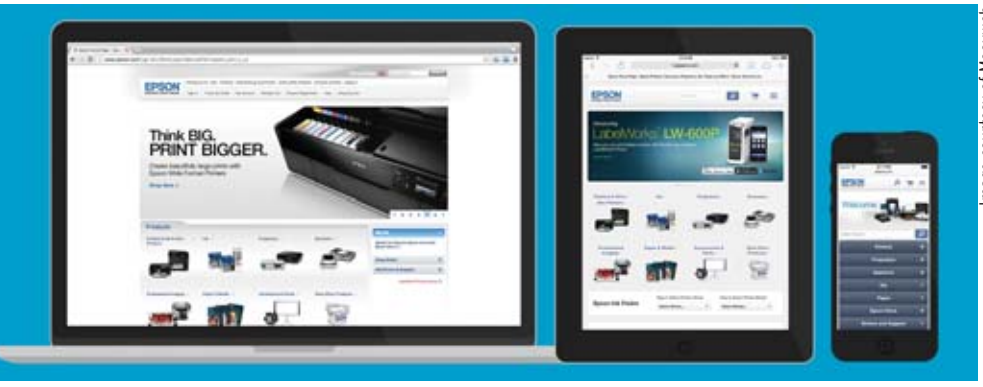


Image courtesy of Moovweb

"Responsive" has simply come to describe techniques and technologies in which a unified set of website code produces Web pages optimized for multiple devices based on screen size and device capability. How a site achieves this unified state is becoming less and less important.

"'Responsive' is coming to mean a single experience across device types," Grannan said. "Whether or not you design or deliver it with real responsive techniques, whether it's responsive, adaptive, RESS or something else, will kind of fade away."

The birth of Responsive Design

On May 25, 2010, Web designer and developer Ethan Marcotte published the seminal article, "Responsive Web Design," in the online magazine A List Apart, followed by a 2011 book of the same name. He was inspired by the concept of responsive architecture, where a building or structure responds and adapts to its environment.

"Rather than tailoring disconnected designs to each of an ever-increasing number of web devices, we can treat them as facets of the same experience," Marcotte wrote. "We can design for an optimal viewing experience, but embed standards-based technologies into our designs to make them not only more flexible, but more adaptive to the media that renders them. In short, we need to practice *responsive web design* [emphasis Marcotte's]."

The precise technical definition of RWD is made up of three components: fluid grids, flexible images and media queries. The fluid grids set the width of the Web page based on the dimensions of the device, and, along with the flexible images, automatically adjust and resize based on a device's screen dimensions when the code unpacks.

"Imagine a three-column layout for a newspaper," said Ishan Anand, director of new products at cloud-based responsive platform provider Moovweb. "As you continue to shrink that layout smaller and smaller, it doesn't make sense on a mobile phone to continue to have three columns, even if each of those columns are getting proportionally smaller. At a certain point, it breaks down."

The third component, a CSS3 feature called a media query, is how the browser identifies its own width. At a certain number of pixels, the media query will identify a device as a tablet or a smartphone and that three-column newspaper layout will consolidate to two or one, reorganizing the content accordingly.

"So you have HTML, CSS and JavaScript as the three pieces that make up a Web page," Anand said. "With Responsive Web Design, you keep the same HTML and you're just changing the CSS, but that fundamentally limits your design choices. Sometimes when you're trying to create a design or an experience for a Web page, you need to actually change the structure of the HTML."

The growing umbrella of 'responsive'

As "responsive" has developed more into an overarching concept than a particular methodology, other techniques have emerged to fill in the gaps where RWD falls short. The earliest example of this is Adaptive Web Design (AWD), popularized in the 2011 book "Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement," by Aaron Gustafson.

AWD flips the single client-side template approach of RWD on its head. It instead uses a different template for each device, which the server identifies before sending the appropriate template to the client. AWD solves the issue of performance by sending only the code a particular device needs, rather then overloading a smartphone with an entire responsive codebase, including desktop and tablet specifications it won't need.

"Adaptive design leverages different design back ends so your Web server has all the different experiences, and based on what you can tell about the device when users access the site, you deliver up a specific subset of context," Forrester's Grannan said. "The site can tell if I'm coming from an Android versus a desktop browser."

Another offshoot of RWD is RESS, a technique laid out in a 2011 article by "Mobile First" author Luke Wroblewski, which blends responsive and adaptive techniques. Using RESS, the page is still constructed according to RWD principles, but individual components within the page can be optimized for a specific device by server-side code.

"RESS was designed to deal with these problems of performance that Responsive Web Design was having," Moovweb's Anand explained. "With Responsive Web Design, the mobile device is getting the same code as the

desktop, even though they're radically different in network access, memory, CPU and processing power. RESS divides the page into components, and those components are placed together according to Responsive Web Design, but the server might decide to substitute those components out depending on the device."

On the "responsive" spectrum, RWD sits on one end with its client-side universal template, AWD sits at the other with multiple sever-side templates, and RESS lands in the middle. The server-side properties of AWD and RESS both lend themselves to more customization of the user interface of each device, addressing another limitation of RWD.

Outside of that spectrum, another technique similar to RESS arose to tackle the time- and labor-intensive back-end development process for enterprises. Forrester's 2014 Enterprise Mobility Survey of 146 U.S. enterprise companies, commissioned by Moovweb, found that more than 70% of cost, time and labor on RWD projects are spent on the back-end re-coding of APIs, middleware and infrastructure.

Responsive Delivery (RD), a technique defined and shepherded into the marketplace by Moovweb, transforms an enterprise website's existing code into mobile HTML5, but does so in a cloud-based platform rather than on the server-side.

"What Responsive Delivery does, in a nutshell, is it moves [code] transformations to the cloud before they hit the device," Anand said. "Responsive Delivery can be applied to a site that wasn't designed with mobile in mind. To make your website responsive typically involves a rewrite of the entire site, and responsive rewrites are notorious for taking months or years. To solve those problems, Responsive Delivery just defines a series of transforms to transition the existing site into a responsive one."

The future of RWD

"Responsive Web Design gets a lot of the credit for catalyzing the discussion around mobility and the right way to go mobile, but invariably we need to refine it," Anand said. The broadening of the term is a reflection of that. In any of those techniques, responsive delivery, Adaptive Web Design or RESS, you're still incorporating some of the principles of Responsive Web Design as a starting point, but you're building upon it to improve it."

As the boundaries of what "responsive" means have shifted, the concept has begun to move beyond any particular technique. Responsive elements are even finding their way into development tools, as companies such as developer tools vendor Telerik is integrating responsive elements, both responsive and adaptive, to its Kendo UI widgets, implementing flexible layouts and media queries for simpler widgets, and new mobile templates for more complex widgets.

According to Telerik director of product management Brandon Satrom, the decision to go "responsive" was to give developers more flexibility in creating apps or websites for whatever smart device or screen size comes next.

"What if somebody puts a browser on a Galaxy Gear watch?" said Satrom. "What do I do about that? What does it mean that there's a browser on my Xbox, and how do I support the features and functionality and the fact that somebody may be looking at this on a 65-inch television? You can't keep a list anymore of all the different screen sizes and form factors and device pixel ratios. You have to start from a position of flexibility."

RWD is the first step in a still-evolving pursuit of creating optimized environments for an ever-expanding range of devices. Forrester's Grannan sees responsive as a mindset going forward for both understanding and unifying a user's experience.

"From an evolutionary standpoint, I'd say we're absolutely moving away from Ethan Marcotte's pure vision of Responsive Web Design," he said. "We're moving more toward a broader understanding of multi-device and multi-channel experiences, and making those not homogenous but coherent across the spectrum of digital touch points that customers are digesting." ▮

Read this story on sdtimes.com

Responsive Design in dedicated mobile websites

When it comes to mobile development, RWD has traditionally been at odds with the dedicated mobile website, a separate entity from the desktop site usually discernable by the "m" in front of the URL. Yet as the term "responsive" has broadened, so too have possible applications of responsive elements.

The conventional thinking behind choosing a mobile website over RWD is that a dedicated mobile site can handle the performance load of a transactional site such as Facebook.com or Amazon.com, or the data and content-heavy load of a site like ESPN.com. RWD is viewed as better suited to simple public websites or publishing sites where the user is consuming a single workload of content.

Joe Herres, executive vice president of products at SharePoint consulting and mobile product company H3 Solutions, has been outspoken about the limitations of RWD. Yet he sees the possible benefits of incorporating responsive elements, such as a single URL and adaptive templates, into dedicated mobile sites.

"Responsive Design is not the end-all be-all solution of multi-channel delivery of the Web, but it's useful to implement into any website or Web design," Herres said. "I think it's really consolidating the user's experience. What's great about responsive is it's the same URLs, the same place you're going to, but it's rendered differently for different form factors.

"Going forward we may see more and more mobile websites using Adaptive Web Design," he said. "I wouldn't doubt in the future when I go to Facebook.com on my iPhone, I'm no longer going to see that 'm.facebook.com.' It's still going to be 'www.facebook.com,' but it's going to be adaptive. It's going to send you mobile-specific renderings. I think we're moving more into a combination of adaptive and responsive, so that each respective form factor will receive the premium experience without having to degrade for the other." ▮

—Rob Marvin

Code Watch

BY LARRY O'BRIEN

Larry O'Brien, former Editor of Software Development and Computer Language magazines, is a software developer living in Hawaii.

Embrace Java 8 (with some caution)

Java 8 has arrived. Every Java shop should embrace this new language version, which represents the most significant advance in the language since the 1990s. The headline feature is the addition of lambdas: in-line anonymous functions and important touchstones on the path toward the functional programming paradigm.

The availability of lambda functions is a low threshold to qualify a language as "functional," and Java 8 is the last of the mainstream languages to add support for first-class functions. In that strict sense, we're all functional programmers now. And, as I've said, certain functional techniques and mindsets have become commonplace. Unit testing in particular has shown us all that it's easier to maintain a set of smaller functions that each compute and return a single thing rather than a monolithic function that does a complex calculation and then stores the result as internal state.

Many development managers are rightfully conservative, and they recognize the risk of introducing newly learned techniques into mission-critical codebases. This is wise when it comes to functional programming. Although I believe that functional programming techniques are broadly beneficial, a study by Pankratius, Schmidt and Garretón showed that "Scala code is more compact than Java code, but clearly refute other claims of Scala on lower programming effort and lower debugging effort."

I always caution about the difficulty of generalizing from specific experiments, but I'm going to ignore my own advice because this conclusion agrees with my own experience in using Scala. I think Scala is an excellent language, and would choose it over Java—even Java 8—for new development by a small, experienced team. With larger codebases and situations where maintenance programming is already a significant part of the budget, choosing Scala has the downside that there are relatively few Scala programmers, and those programmers are expensive.

Java, on the other hand, is the most popular of the mainstream languages for corporate development. The downside of that is that the population of Java programmers contains many developers who have not been pushing themselves to continue learning, and who may be overconfident about their ability or blind to the challenges of functional code.

Functional code can be very dense, especially with functions that manipulate other functions. It can also be abused into a mess of spaghetti (when lambdas define lambdas defining lambdas, etc.). Object-oriented and functional designs do not always merge cohesively, especially in larger codebases where sweeping evolution to an entire module is not practical. Dense code, spaghetti code and confusing designs are not unique to functional programming, of course, but they are areas where developers who don't know better can dig themselves into deeper and deeper holes.

Compounding the problem is the pragmatic issue that debuggers haven't evolved to deal well with first-class functions. This holds true for the Java 8 preview IDEs that I have looked at. Debuggers have become excellent at allowing developers to inspect values, but I've yet to see one that excels at visualizing a variable that is actually a handle to a function or that is focused on a program whose state is primarily stored in the stack, not the heap.

Java 8's Stream library will gave Java shops similar power to the kind enjoyed by C# developers using LINQ.

Having raised a few caution flags, I want to reiterate my belief that Java 8 should be embraced. I've never met a developer who doesn't rapidly grow to appreciate the elegance of using higher-order functions for transforming collections, and Java 8's Stream library will give Java shops similar power to the kind enjoyed by C# developers using LINQ. This power includes not only querying and filtering, but also a firmer foundation for reactive programming.

Many would say that, even more than first-class functions, the essence of functional programming stems from the immutability of variables after they have been assigned, and lazy evaluation—the ability to "call by need" in a manner that allows for such things as infinite sequences and data structures and flexible control-flow manipulation. Although I was initially hopeful that the Stream library was based on lazy evaluation, it appears not, but third-party libraries have already begun to appear.

Finally, I want to praise Java 8's "default methods," which are interfaces with implementations. It facilitates the Data-Context-Interaction approach, which I very much like.

Java 8 is a 450MB download from Oracle. Best of all, it doesn't include the Ask toolbar. ■

Read this story on sdtimes.com

Geoffrey Vaughan is an IT security consultant at Security Compass, an information security firm.

Guest View

BY GEOFFREY VAUGHAN

Threats magnified in HTML5

By now, developers have grown accustomed to dealing with certain "traditional" threats when it comes to Web applications. For instance, issues like injection flaws and cross-site scripting have been on the OWASP Top 10 list for years. But take those same Web applications and convert them to a mobile HTML5 application, and suddenly older and "lower-impact" threats become significantly more high-risk for the mobile platform.

It's important for developers to remember that any vulnerability that affects Web applications can also affect mobile HTML5 applications. And because of an increased availability of features, including local storage on the device, access to contacts, location, and other sensitive information with mobile apps, these threats can become magnified in a mobile environment.

Because of an increased availability of features in HTML5 apps, threats can become magnified.

Adding to this problem is that many developers use multi-platform development/deployment tools that can introduce vulnerabilities into the application. Not to mention, many Web app specialists are now being asked to develop mobile apps without a proper understanding of security implications.

Here are five traditional threats that are magnified in a mobile HTML5 environment:

1. **Local Storage:** With traditional Web apps, local storage is mostly a moot point (with the exception of cookies), so the risk is only minimal. But in a mobile app, local storage is key to the user experience, and this practice elevates the impact of this vulnerability.

All too often, mobile HTML5 apps leave sensitive user data throughout the user's memory space that can be accessed by an unauthorized user. This may include plaintext files, improperly encrypted files (such as unsalted hashes), or even the encryption keys themselves. Developers must assume that any data put into the user's memory space will be seen and manipulated. **Tip for Developers:** Do not store any sensitive user data in the local memory space.

2. **Cross-Origin Resource Sharing:** CORS, or the use of both internal and external libraries and resources, is a well-known issue with Web apps, but the risks are often overlooked with mobile HTML5 apps. In fact, these apps are often configured to allow resources to be used and shared among external untrusted sources. This puts the app at a much higher risk of loading malicious external scripts, particularly if hit by a cross-site scripting attack.

While Web apps are also at risk of this, the attack surface is much larger on a mobile app due to its richer features, which makes the damage potentially much greater. **Tip for Developers:** Limit dependence on untrusted code sources and libraries. If this can't be avoided, then the application or server making the request to an external source should be restricted to only use whitelisted sources.

3. **Cross-Site Scripting:** XSS attacks in traditional Web apps can steal the user's session or attack the browser. But with a mobile HTML5 app, the malicious script can also attack device services on the phone (i.e., steal contacts, pictures, whatever else the application has permission to). **Tip for Developers:** Treat all user-inputted data as hostile and untrusted. Data should be filtered to remove any malicious input as well as make use of proper output encoding before any user data is displayed on screen.

4. **SQL Injection:** SQL injection is another common threat for Web apps that has been routinely documented by the OWASP Top 10. But in a mobile HTML5 application, in addition to attacks against a remote server's database, you now also have to worry about rogue applications attacking local databases. **Tip for Developers:** To protect from SQL injections in a local database, in addition to traditional SQL injection defenses (such as using parameterized queries and scrubbing queries for malicious input), avoid storing sensitive user data in local databases.

5. **Cross-Frame Scripting:** Also known as "clickjacking," CFS occurs when a malicious user loads a window or frame on top of a running application, which steals data from the user when tapped or clicked. In mobile HTML5 applications, the impact of CFS is often quite severe as malicious users are sometimes able to manipulate any action on a user's mobile device. **Tip for Developers:** The best way to protect against CFS is to configure the application server so that it only distributes the app if it is in the uppermost view. This server configuration setting is referred to as the X-Frame header options. ▮

Read this story on sdtimes.com

Analyst View

BY ROB ENDERLE

Rob Enderle is a principal analyst at the Enderle Group.

Why you'll be a Borg in 10 years

We've had wristwatch cell phones for some time, and they haven't exactly become a must-have item for anyone. But this was largely because folks wanted to build cheap phones, and the cost of putting a good watch phone on your wrist or another part of your body has been prohibitive.

But step back and think for a moment. If you are like me, you probably have a laptop, a tablet and a smartphone, which is by my count two more things and a level of complexity you'd rather not deal with. If you move the watch to the wrist and integrate the phone capability, you drop one device, and as tablets and laptops continue to drift together, you'll be down to one device you have to carry. Now, if you take a head-mounted high-resolution display and couple it with voice recognition and command, plus a virtual keyboard, you'd be able to give up the tablet and laptop.

Let's talk about that and why it will take a decade or more to get this done.

Smartphones, tablets and laptops

The problem with all three classes of device is that smartphones are too small to type on fast or display as much information as you would like, but they fit in your pocket (at least most do). Tablets are larger but still not large enough for most to live off of, and they are thought to be too large for phone use (though this is clearly changing). Even notebooks don't have as much video real estate as you'd probably like, but they do have a workable, decent-sized keyboard and plenty of performance. But they sacrifice portability and battery life to get there.

A smartphone is best for communicating. A tablet is best for consuming information (text, videos, etc.) because it is still portable. And a laptop is best for creation. The differences are increasingly coming down to two things: Screen size, and how you enter the data (real or virtual keyboard).

Fixing the problem

As head-mounted displays get more and more resolution, and as we figure out how better to blend the real and virtual worlds (by using things like an integrated camera), their utility will grow. And with the right resolution and optics, a head-mounted display can create experiences that rival that of a 40-inch TV.

Voice-to-text capability has been improving sharply, and voice command is already very advanced, which allows you to forgo a keyboard. Further, technologies that allow you to project a keyboard on any service have advanced to the point that folks who want to type without a real keyboard have that option.

The result is that we are very close to having a head-mounted display you could wear all the time, and using voice and a virtual keyboard for different tasks means we no longer require a physical keyboard for data entry.

The eventual solution would constitute a high-resolution head-mounted display with a 4K camera and microphone coupled with a computing unit you'd keep in your pocket or on your belt, or in your purse wirelessly connected to the head-mounted display.

You'd see the world through the display, which would automatically adjust for low light to help you see, and you would be able to bring up information in a heads-up format on your health, location, and businesses around you. It'd also do full PC-level work using a combination of the display and your voice.

It will be increasingly hard to tell what is real and what is virtual, and where the human ends and the machine begins.

You'd interact by talking to the device or using any flat surface as your digital keyboard, and you'd likely be unable to live without the device, which would form an interface between you and both the physical and material worlds.

The Borg future

The Borg were human/machine hybrids created for the "Star Trek" universe and showcased mostly in the series "Star Trek: The Next Generation." With the Borg, it was hard to tell where the machine left off and the human began. The result of having and wearing all of this technology that constantly keeps you connected and filters all that you do and see is that it will be increasingly hard to tell what is real and what is virtual, and where the human leaves off and the machine begins.

What will take time isn't the technology, most of which is cooked. It is our acceptance of living even more connected than we do today, and with looking like rejects from "The Terminator" for most of our lives. ▮

Read this story on sdtimes.com

David Rubinstein is editor-in-chief of SD Times.

Industry Watch

BY DAVID RUBINSTEIN

Be resilient as you PaaS

In the big picture of software development and computing, Platform-as-a-Service is still in its infancy. But because it is tied to "the cloud," people seem to think it's older than it really is.

PaaS confuses people and fills them with fear of vendor lock-in because the early platforms are tied to specific infrastructures. The overwhelming majority of people using the Force.com platform, for instance, are those working with Salesforce's CRM system.

The only alternative up until now has been the private cloud, which allays fears of security that comes up short and of the vendor lock-in issue. Unfortunately, it doesn't scale, and the economic benefits of a true cloud solution have yet to be proven internally.

We'll see an explosion in Platform-as-a-Service when there is standardization in platforms.

So, just where does this leave PaaS going forward? According to Ben Grubin at Boston-based Cloud Technology Partners, a truly effective PaaS is like what the .NET Framework is to C#: You can write the code yourself in C#, or you can take advantage of the broader framework, with the ability to reuse software components between apps and to integrate with outside resources and services. "It becomes a migration target more than what Force.com was," he said. "There's no 'Write your app for MY platform.'"

Amazon is the cloud leader because it has executed this exact strategy perfectly. You can use Amazon Web Services in your applications—which can be reused in other applications—or choose from a host of integrations. "They are offering phenomenal service and they're dominating the market, but not in a way that locks you in," said Grubin.

So, while a number of PaaSes exist, each with their own benefits, there are commonalities among them. They all rest on traditional storage and virtualization. We'll see an explosion in PaaS, Grubin believes, when there is standardization in platforms. This will allow folks to move their applications from one PaaS to another without (significant) modification, and still leverage security and scale. That's an advantage PaaS has over container solutions such as Docker, in which you have to maintain the entire stack to port an application. Also, it only works on Linux for now.

Yet Grubin admits most organizations are nowhere near ready to have this kind of conversation yet. We still haven't seen a tipping point for Infrastructure-as-a-Service, he said, so the tipping point for PaaS is probably another five years out. Companies are still having a hard time getting a handle on what their data center costs even are, what with real estate, hardware, utilities and human resources to be factored in. But they know one thing: They don't want to have to manage those things that are not core to their businesses any longer.

"Enterprises are not talking about cloud adoption. They're talking about data center consolidation. They want to get out of the data center business," Grubin said. "But the biggest problem they have is deciding what to move" out of their data centers and into the cloud.

Millions of enterprise workloads remain in data centers, where servers are 30% to 40% underutilized, and that's if they're virtualized. If not, they're only using 5% to 7% of capacity. "That's a lot of machines making noise, and generating heat, and doing next to nothing," said Grubin. Take, for example, servers that spun up for a project two years ago that were never decommissioned, just sitting there, waiting for a new workload that will never come. And, because the costs of blades and racks went down, cheap hardware has led to a kind of data center sprawl. Now, he said, "It's too big a problem" to untangle.

For developers, a key aspect to writing applications for the cloud is understanding the choice of end target. "Where I write the app has a lot to say about maintainability, and the cost of running it in the long term.

"The future is not resilient hardware; it's resilient software," he continued. "In the old days, you needed hardware resiliency, because when the infrastructure went down, the app died. Today, apps have to be architected to know how to be resilient, so if the server dies, it can run on other nodes."

Developers will need to begin designing for application-layer resiliency once PaaS reaches that tipping point. But it's something to start thinking about now. ▮

Read this story on sdtimes.com

Build Better Software

DevSuite

Quality Integrated ALM for Agile and Non-Agile Projects

Integrate Quality Assurance with Agile Processes

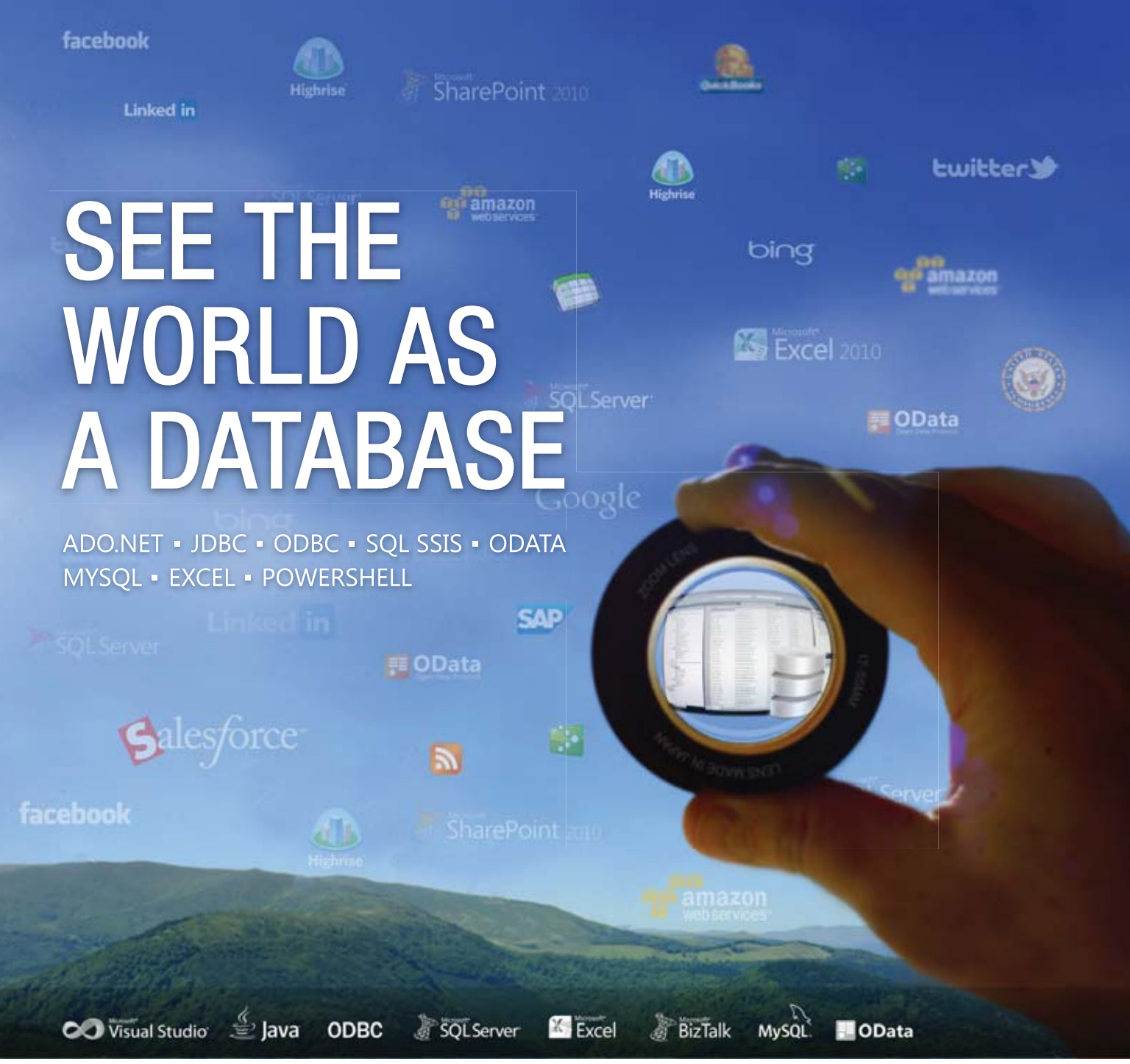
- Define QA test cases in tandem with development cycles
- Set thresholds that ensure quality standards have been met when a "Shippable Product" is produced
- Integrate automated and manual testing with agile development

Test Management and Defect Tracking for Traditional Projects

- Define QA test cases and a QA test workflow
- Automatic defect submission from failed test cases
- Quantify product quality so you can objectively measure the quality of a release

TechExcel

(800) 439-7782 | www.techexcel.com



Work With Relational Data, Not Complex APIs or Services

Whether you are a developer using ADO.NET, JDBC, OData, or MySQL, or a systems integrator working with SQL Server or Biztalk, or even an information worker familiar with ODBC or Excel - our products give you bi-directional access to live data through easy-to-use technologies that you are already familiar with. If you can connect to a database, then you will already know how to connect to Salesforce, SAP, SharePoint, Dynamics CRM, Google Apps, QuickBooks, and much more!

Give RSSBus a try today and see what we mean:

visit us online at www.rssbus.com to learn more or download a free trial.

rssbus

INTEGRATION YOUR WAY

Attend the Largest Dedicated Android Development Conference in the Universe!

PRICE OF CONFERENCE REDUCED!
Click here for details

AnDevCon
May 27-30, 2014
Sheraton Boston

Check out listing of
CLASSES AND TUTORIALS INSIDE!

Get the best real-world Android developer training anywhere!

- Choose from more than 75 classes and in-depth tutorials
- Network with speakers and other Android developers
- Check out more than 40 exhibiting companies

Keynotes Announced!

YAHOO!

intel Software

QUALCOMM

Check online for details!

DIAMOND SPONSOR

intel Software

PLATINUM SPONSORS

BlackBerry

QUALCOMM

CENTRAL CONTRACTOR REGISTRATION

Register Early and SAVE at www.AnDevCon.com

A BZ Media Event #AnDevCon



Check out this list of TUTORIALS!

Who's Afraid of ContentValues?
Ian Darwin

Working with Video on Android
Rory Hool

Android Networking for the Enterprise
G. Blake Meike

Embedded Android Tutorial
Karim Yaghmour

Understanding Android Studio and its Role in the Android Developer Toolset
Jim Wilson

FOR FULL DESCRIPTIONS OF SESSIONS
Go to AnDevCon.com

A Totally Different View: Under the Hood with Android and Linux
Jonathan Levin

Android Application Development for .NET Developers
Michael Crump

Encrypt all Things
Nathan Freitas

How to Test Apps with Appium
Brian Kitchener

Leveraging Fully Customized ViewGroups From the Ground Up
James Baca

OpenGL-ES 3.0 and Beyond—Modern OpenGL Programming
Ron Fosner



Check out this list of CLASSES!

A Deep Dive into Renderscript
Larry Schiefer

An Introduction to Building Enterprise-Secure Android Apps
G. Blake Meike

Analytics Theory and Practice
Nathan Mellor

Android as the New Standard Embedded OS
Karim Yaghmour

Android Emulator Myths... Busted
Luis de la Rosa

Android Platform Debugging and Development
Karim Yaghmour

Android Security: New Threats, New Capabilities
Jonathan Levin

Apache Maven for Android Development Pro
Manfred Moser

App Discovery Optimization
Nathan Mellor

Apps For Profit
Nathan Mellor

Automotive Android
Godfrey Nolan

Battle-Tested Patterns in Android Concurrency, Parts I and II
Doug Stevenson

Battling the Media Framework
Brad Grimm

Bootstrapping Android Development
Donn Felker

Building Android for the Cloudstarburst
Ron Munitz

Cloud-Powered Android Apps in Minutesstarburst
Chris Risner

Connect Your App to the Real World with Google Services Location API
Jim Wilson

Continuous Integration Server Usage for Android Development in Practice
Manfred Moser

Creating Composite Views in Android
James Harmon

Customizing Android for Fun and Profit, Parts I and II
Dave Smith

Designing for Android
Chris Huban and Vijay Penemetsa

more on next page!

MORE CLASSES!

Developing Android Bluetooth Smart Ready Apps
Vincent Gao

Dive into Android Fragments
Donn Felker

Dynamic Audio for Apps and Games
Tony Hillerson

Five Strategies to Achieve Continuous Unattended Testing
Uzi Eilon

Git for the Android Developer, Parts I and II
Gradle: From User to Addict
Jake Ouellette

Home Sweet Home: Widgets and Live Wallpapers
Elizabeth Mezias

How to Maximize the Power Usage and Performance of your Apps
Rick Schwartz

In-depth OAuth for Android
Luis de la Rosa

Inside Android's User Interface
Karim Yaghmour

Intellectual Property and Apps, Parts I and II
Vlad Shvartsman

Localizing Your Apps: Why? How? Parts I and IIs
Matt Brenner

Mastering Notifications in Android Applications
Travis Himes

Measuring and Improving Your App's Network Performance
Doug Sillars

Multi-User Android: The Complete Guide
Ron Munitz

NDK Primer
Ron Munitz

Push Message Primer
Mark Wickham

Pushing to Android with Azure
Chris Risner

RxJava on Android
Dan Osipov

Putting Your App on a Memory Diet, Parts I and II
Mark Murphy

Practical Build Automation: Embracing Gradle
Avram Lyon

Preparing your APIs for the Future
Ole Lensmar

Say Cheese: Building a Custom Camera Application
Huyen Tue Dao

Stacking Content Providers
Ian Darwin

Tactics and Tools from a Professional Android Team
Derek Brameyer

FOR FULL DESCRIPTIONS OF SESSIONS
Go to AnDevCon.com

Sponsors and Exhibitors

DIAMOND SPONSOR

PLATINUM SPONSORS

GOLD SPONSORS

SILVER SPONSORS

OTHER EXHIBITING COMPANIES

More exhibitors being added every day! Check online for updates!

Produced by
BZ Media
Android Developer News

Event Schedule

Tuesday, May 27

7:30 am – 7:00 pm	Registration Open
8:00 am – 9:00 am	Morning Coffee
9:00 am – 10:30 am	Tutorials
10:30 am – 10:45 am	Coffee Break
10:45 am – 12:30 pm	Tutorials
12:30 pm – 1:30 pm	Lunch
1:30 pm – 3:00 pm	Tutorials
3:00 pm – 3:15 pm	Coffee Break
3:15 pm – 5:00 pm	Tutorials

Wednesday, May 28

7:30 am – 7:00 pm	Registration Open
7:30 am – 8:45 am	Morning Coffee
8:45 am – 9:45 am	Keynote – Yahoo
9:45 am – 10:00 am	Coffee Break
10:00 am – 11:15 am	Technical Classes
11:30 am – 12:45 pm	Technical Classes
12:45 pm – 1:45 pm	Lunch
1:45 pm – 2:15 pm	Sponsored Sessions
2:30 pm – 3:45 pm	Technical Sessions
3:45 pm – 4:00 pm	Coffee Break
4:00 pm – 5:15 pm	Technical Classes
5:30 pm – 7:00 pm	Lightning Talks
7:30 pm – 9:30 pm	Android Design Panel: Design Is The New Black

Thursday, May 29

7:30 am – 7:00 pm	Registration Open
7:30 am – 8:30 am	Morning Coffee
9:00 am – 10:15 am	Technical Classes
10:30 am – 11:00 am	Keynote – Intel
11:00 am – 11:30 am	Coffee Break
11:00 am – 7:00 pm	Exhibit Hall Open
11:30 am – 12:00 pm	Sponsored Sessions
12:00 pm – 1:15 pm	Lunch
12:00 pm – 1:15 pm	Women in Android Luncheon
1:15 pm – 2:30 pm	Technical Classes
2:30 pm – 3:00 pm	Coffee & Ice Cream Break
3:00 pm – 3:30 pm	Keynote – Qualcomm
3:45 pm – 5:00 pm	Technical Classes
5:15 pm – 5:45 pm	Keynote
5:45 pm – 7:30 pm	Networking Reception

Friday, May 30

7:30 am – 4:45 pm	Registration Open
7:30 am – 8:30 am	Morning Coffee
8:30 am – 9:45 am	Technical Classes
9:45 am – 10:00 am	Coffee Break
10:00 am – 11:15 am	Technical Classes
11:00 am – 2:30 pm	Exhibit Hall Open
11:00 am – 11:30 am	Coffee Break
11:30 am – 12:00 pm	Sponsored Sessions
12:15 pm – 1:15 pm	Lunch
1:15 pm – 2:00 pm	Winner's Circle Prizes Announced
2:00 pm – 3:15 pm	Technical Classes
3:15 pm – 3:30 pm	Coffee Break
3:30 pm – 4:45 pm	Technical Classes
4:45 pm	Conference Closes

Getting Approval

Gotta Get Approval?
Try These 10 Time-Tested Tactics:

1. STUDY. Find the classes and tutorials at the Android Developer Conference focused on Developer Essentials, Android Business, Tablets and Beyond, and Embedded Android. Find the sessions that are best for you.

2. PREPARE. Download the course catalog and circle the classes you want to take, and explain why the topics relate to your Android technical efforts.

3. CHOOSE. There are many sessions offered in each time slot. That means that you'll always find something that fits your needs and is at just the right level for your own Android development and management needs.

4. JUSTIFY. Go to your manager armed with all the necessary materials to make a good case for how AnDevCon will help your company make money, save money and improve productivity.

5. SHARE. Promise to come back from AnDevCon and hold a brown-bag lunch session to share what you've learned with your colleagues, or even conduct formal training within your department.

6. SAVE. The sooner you register, the more your company saves, so explain the benefit of signing up early, both for the conference and for the hotel, and for making your travel arrangements right away.

7. TEAM. Save even more with group discounts. Send three or more employees from your company and save $100 per person. Each person can take different classes and bring back even more valuable tips and techniques.

8. BIG TEAM. Are 10 or more of you coming to AnDevCon? Contact Stacy Burris for special arrangements, at sburris@bzmedia.com.

9. DISCOUNTS. User groups, government employees, non-profits and professionals employed by or attending educational institutions are eligible for special savings.

10. ACT. While you can sign up anytime, your company will save the most if you beat the early deadlines. Help your company's bottom line by signing up today!

Find out why you should go to AnDevCon! Watch the videos at www.AnDevCon.com

Hotel and Travel

AnDevCon Boston 2014 will be held at the Sheraton Boston Hotel.

Sheraton Boston Hotel
39 Dalton Street
Boston, Massachusetts, USA 02199
1-617-236-2000
1-800-325-3535

Reservations

Special Discounted Rates
Take advantage of special discounted room rates at the Sheraton Boston Hotel — only US$209 per night for single/double occupancy. Rooms for the reduced rate are limited!

Reservations at the reduced rate can be made through 5:00 pm Eastern time on May 13, 2014 — assuming they don't sell out. The number of rooms in the discounted block is LIMITED and historically rooms sell out well before the deadline. Don't wait until the last minute to reserve your hotel rooms!

This rate is available throughout the duration of the AnDevCon conference. Also, if you reserve your room via the registration link you will receive free wireless Internet service in your room.

Current parking rates
0-1 hour: $16
1-2 hours: $22
2-4 hours: $28
4-8 hours: $31
Overnight: $46

After 8 hours, overnight rate applies. Registered guests must remove vehicle by 2pm on day of checkout or additional parking charges may be levied.

Due to the limited space and height restrictions of the parking facility, oversized vehicles will be charged $88 per night and oversized parking is at a first come first serve basis. Any vehicle exceeding a height of 6' and/or exceeding 17' in length is considered oversized.

Hertz Rental Car Services
Whether you are looking to take a day trip up to Vermont or down Cape Cod, or may have arrived into Boston via nearby Back Bay Station; we offer all guests the convenient option to rent a car directly in our lobby. A fully staffed Hertz rental desk is located adjacent to the gift shop.

To reach the direct Hertz desk in our lobby, call 617-267-0395 or to reach Hertz main toll-free number, call 800-654-3131.

Trying to figure out how to get to the Sheraton? View directions to the Sheraton Boston.

AnDevCon is definitely worth the trip. It's the best place to learn about Android, and a great way to jumpstart your skills for those getting started.
—Rory Hool, Software Engineer, Tech Smith

Registration

Register TODAY at www.AnDevCon.com!

	Register By May 16	Register On or After May 17
Three-Day Conference PLUS Pre-Conference Tutorials May 27, 28, 29, 30, 2014	**$1,195** SAVE $300	**$1,495**
Three-Day Conference May 28, 29, 30, 2014	$995	$1,295
Pre-Conference Tutorials Only May 27, 2014	$745	$795
Exhibit Hall Only Pass May 29, 30, 2014	FREE	FREE

PRICE REDUCED!

All fees are in US$

An excellent conference with access to a lot of brilliant speakers.
— Mandalapa, Web Developer, Viacom Inc.

Three-Day Conference PLUS Pre-Conference Tutorials Registration Includes:

- Admission to pre-conference tutorials on May 27
- Admission to sessions and technical classes on May 28, 29, 30
- Admission to keynotes
- Admission to Exhibit Hall
- Admission to all special events, including the Networking Reception
- Conference materials – PDF files available for download on web site
- Coffee breaks and lunch where indicated
- Certificate of Completion

Three-Day Conference Registration Includes:

- Admission to sessions and technical classes on May 28, 29, 30
- Admission to keynotes
- Admission to Exhibit Hall
- Admission to all special events, including the Networking Reception
- Downloadable conference materials
- Coffee breaks and lunch where indicated
- Certificate of Completion

Pre-Conference Tutorials Only Registration Includes:

- Admission to pre-conference tutorials on May 27
- Downloadable conference materials
- Coffee breaks and lunch where indicated
- Certificate of Completion

Exhibit Hall Pass Registration Includes:

- Admission to the Exhibit Hall
- Admission to Networking Reception and all other special events

HOW TO REGISTER

Register online and use one of the following payment methods:

Credit Card. You can use the secure online form to pay via credit card and get immediate confirmation of your registration. MasterCard, Visa and American Express are accepted. You'll receive a registration record and receipt. Please print out these pages and bring them with you to the Conference. Present them at the Registration Desk to pick up your badge and course materials.

Check. Fill out the online registration form. Print out the registration record and receipt, then mail to BZ Media LLC, 225 Broadhollow Road, Suite 211, Melville, NY 11747 with your payment. Online registrations that are mailed without payment will not be confirmed until payment is received.

Purchase Order. If you register using a P.O., you'll be invoiced immediately for the registration amount. Payment must be received before your registration can be confirmed.

CANCELLATION AND REFUND POLICY

You can receive a full refund, less a $150 registration fee, for cancellations made by Friday, April 18, 2014. Cancellations after this date are non-refundable. Send your cancellation in writing to registration@bzmedia.com. Registrations may be transferred to another person.

Refunds will be processed through the same method of payment as the initial payment transaction. Credit card refunds will be processed to the same credit card as the original payment.

If for reasons beyond our control the conference cannot take place as scheduled, BZ Media reserves the right to reschedule the conference to a date and place of its choosing.

QUESTIONS

Contact Stacy Burris, Event Director at sburris@bzmedia.com or +1-631-421-4158 x108.

SPECIAL DISCOUNTS

You may combine one of these special discounts with the Early Registration pricing to save even more!

Group. Group discounts will be given automatically if you register three or more people at once. You can also contact Stacy Burris at sburris@bzmedia.com to receive the $100/person discount if your group is unable to register at the same time. Contact her also for special discounts for groups of 10 or more.

Government Employees. Federal, State and Local Government employees can receive an additional $100 off the Full Event Passport PLUS Pre-Conference Tutorial price. Enter code GOV in discount code field. CCR-registered indicates that we are listed in the primary supplier database for the Federal Government.

CENTRAL CONTRACTOR REGISTRATION

Educational Institutions. Personnel employed by or attending educational institutions can get a $100 discount off the Full Event Passport PLUS Pre-Conference Tutorial price by using the code EDU.

User Groups. Contact Stacy Burris at sburris@bzmedia.com to see if your group is eligible for a discount.

Non-Profit Organizations. Personnel employed by non-profit organizations can get a $100 discount off the Full Event Passport PLUS Pre-Conference Tutorial price by using the code NONPROFIT.

043014